Priority queues and pattern classes

Mike Atkinson

PP 2012
13 June 2012
Joint work with Michael Albert

Outline of talk

New pattern classes from old

- We know many "constructions" that transform pattern classes into pattern classes.
  - $\mathcal{X} \longrightarrow \mathcal{X} \oplus \mathcal{C}$ for some fixed pattern class $\mathcal{C}$.
  - $\mathcal{X} \longrightarrow \mathcal{X}\mathcal{C}$ for some fixed pattern class $\mathcal{C}$.
  - $\mathcal{X} \longrightarrow wk(\mathcal{X})$ (weak closure of $\mathcal{X}$).
  - lots more.
- Given any such construction $\mathcal{X} \longrightarrow \mathcal{X}^*$ we try to find methods to deduce properties of $\mathcal{X}^*$ from properties of $\mathcal{X}$.
- So any construction allows us to incrementally explore the domain of pattern classes

New pattern classes from old

- We know many "constructions" that transform pattern classes into pattern classes.
  - $\mathcal{X} \longrightarrow \mathcal{X} \oplus \mathcal{C}$ for some fixed pattern class $\mathcal{C}$.
  - $\mathcal{X} \longrightarrow \mathcal{X}\mathcal{C}$ for some fixed pattern class $\mathcal{C}$.
  - $\mathcal{X} \longrightarrow wk(\mathcal{X})$ (weak closure of $\mathcal{X}$).
  - lots more.
- Given any such construction $\mathcal{X} \longrightarrow \mathcal{X}^*$ we try to find methods to deduce properties of $\mathcal{X}^*$ from properties of $\mathcal{X}$.
- So any construction allows us to incrementally explore the domain of pattern classes

Uncountably many constructions

- Generalize the pattern class order to an order on pairs of permutations.
- Consider the pair $(32415, 23145)$
    - ... and any subset of the elements e.g. $\{1, 3, 4\}$
    - So $(32415, 23145)$
    - Pick out the pair $(341, 314)$
    - and relabel to get the pair $(231, 213)$
    - Then $(231, 213) \preceq (32415, 23145)$
- This is a partial order on the set of pairs of permutations.

Uncountably many constructions

- Generalize the pattern class order to an order on pairs of permutations.
- Consider the pair $(32415, 23145)$
  - …and any subset of the elements e.g. $\{1, 3, 4\}$
  - So $(32415, 23145)$
  - Pick out the pair $(341, 314)$
  - and relabel to get the pair $(231, 213)$
  - Then $(231, 213) \leq (32415, 23145)$
- This is a partial order on the set of pairs of permutations.

Uncountably many constructions

- Generalize the pattern class order to an order on pairs of permutations.
- Consider the pair $(32415, 23145)$
  - ... and any subset of the elements e.g. $\{1, 3, 4\}$
  - So $(32415, 23145)$
  - Pick out the pair $(341, 314)$
  - and relabel to get the pair $(231, 213)$
  - Then $(231, 213) \leq (32415, 23145)$
- This is a partial order on the set of pairs of permutations.

Uncountably many constructions

- Generalize the pattern class order to an order on pairs of permutations.
- Consider the pair $(32415, 23145)$
  - …and any subset of the elements e.g. $\{1, 3, 4\}$
  - So $(32415, 23145)$
  - Pick out the pair $(341, 314)$
  - and relabel to get the pair $(231, 213)$
  - Then $(231, 213) \leq (32415, 23145)$
- This is a partial order on the set of pairs of permutations.

Uncountably many constructions

- Generalize the pattern class order to an order on pairs of permutations.
- Consider the pair $(32415, 23145)$
  - …and any subset of the elements e.g. $\{1, 3, 4\}$
  - So $(32415, 23145)$
  - Pick out the pair $(341, 314)$
  - and relabel to get the pair $(231, 213)$
  - Then $(231, 213) \leq (32415, 23145)$
- This is a partial order on the set of pairs of permutations.

Uncountably many constructions

- Generalize the pattern class order to an order on pairs of permutations.
- Consider the pair $(32415, 23145)$
  - ... and any subset of the elements e.g. $\{1, 3, 4\}$
  - So $(32415, 23145)$
  - Pick out the pair $(341, 314)$
  - and relabel to get the pair $(231, 213)$
  - Then $(231, 213) \leq (32415, 23145)$
- This is a partial order on the set of pairs of permutations.

Uncountably many constructions

Definition

A *pair pattern class* is a down set in the order on pairs.

- There are uncountably many pair pattern classes and very little is known about them.
- Pairs of permutations are the natural generalization of permutations from 2 dimensions to 3 dimensions.

Theorem

Let $\mathcal{R}$ be any pair pattern class. For every pattern class $\mathcal{X}$

$$\mathcal{X}\mathcal{R} = \{\tau \mid (\sigma, \tau) \in \mathcal{R} \text{ for some } \sigma \in \mathcal{X}\}$$

is also a pattern class.

Uncountably many constructions

Definition

A *pair pattern class* is a down set in the order on pairs.

- There are uncountably many pair pattern classes and very little is known about them.
- Pairs of permutations are the natural generalization of permutations from 2 dimensions to 3 dimensions.

Theorem

Let $\mathcal{R}$ be any pair pattern class. For every pattern class $\mathcal{X}$

$$\mathcal{X}\mathcal{R} = \{\tau \mid (\sigma, \tau) \in \mathcal{R} \text{ for some } \sigma \in \mathcal{X}\}$$

is also a pattern class.

Uncountably many constructions

Definition

A *pair pattern class* is a down set in the order on pairs.

- There are uncountably many pair pattern classes and very little is known about them.
- Pairs of permutations are the natural generalization of permutations from 2 dimensions to 3 dimensions.

Theorem

Let $\mathcal{R}$ be any pair pattern class. For every pattern class $\mathcal{X}$

$$\mathcal{X}\mathcal{R} = \{\tau \mid (\sigma, \tau) \in \mathcal{R} \text{ for some } \sigma \in \mathcal{X}\}$$

is also a pattern class.

What is a priority queue?

A priority queue is a container which can contain priorities (or data items which have a priority). There are two main operations possible on the container:

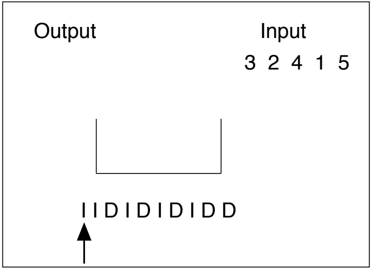
Insert Insert a new item into the container

Delete-Min Delete the item of smallest priority from the container

Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

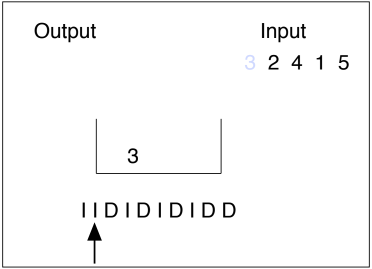
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

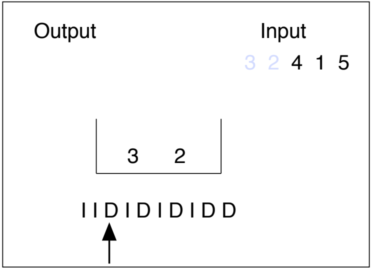
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

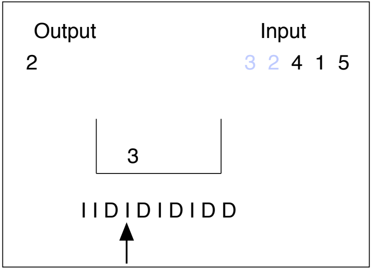
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

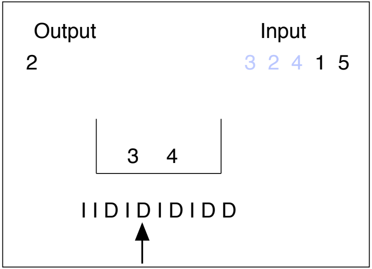
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

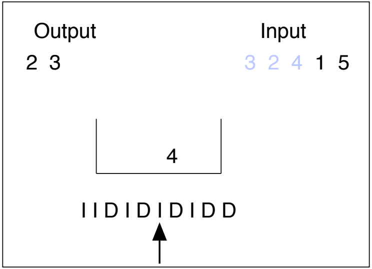
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

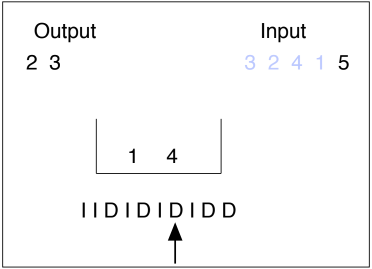
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

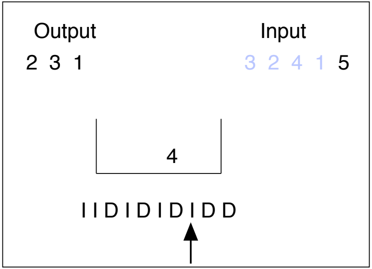
A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

It takes a sequence of input items and produces a sequence of output items



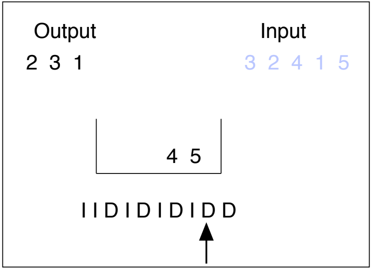
Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

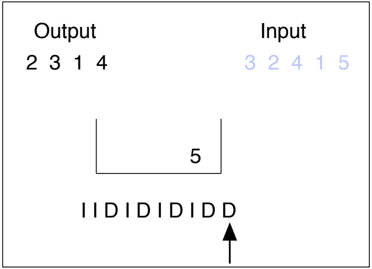
A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

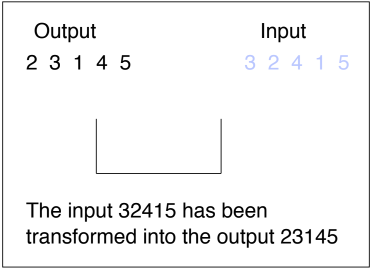
It takes a sequence of input items and produces a sequence of output items



Priority queue computations

A priority queue computation is a sequence of Insert (I) and Delete-Min (D) operations that begins and ends with the priority queue in the empty state such as $IIDIIDIDDD$.

It takes a sequence of input items and produces a sequence of output items



The input 32415 has been transformed into the output 23145

Allowable pairs

- We have just seen a priority queue computation that transformed the input sequence 32415 into the output sequence 23145.
- $(32415, 23145)$ is an *allowable pair*
- The set $\mathcal{A}$ of allowable pairs is a pair pattern class
- There are $(n+1)^{n-1}$ allowable pairs of length n
- The basis of $\mathcal{A}$ is $\{(12, 21), (321, 132)\}$

Allowable pairs

- We have just seen a priority queue computation that transformed the input sequence 32415 into the output sequence 23145.
- $(32415, 23145)$ is an *allowable pair*
- The set $\mathcal{A}$ of allowable pairs is a pair pattern class
- There are $(n+1)^{n-1}$ allowable pairs of length n
- The basis of $\mathcal{A}$ is $\{(12, 21), (321, 132)\}$

Allowable pairs

- We have just seen a priority queue computation that transformed the input sequence 32415 into the output sequence 23145.
- $(32415, 23145)$ is an *allowable pair*
- The set $\mathcal{A}$ of allowable pairs is a pair pattern class
- There are $(n+1)^{n-1}$ allowable pairs of length n
- The basis of $\mathcal{A}$ is $\{(12, 21), (321, 132)\}$

Restatement of previous theorem

Theorem

If a priority queue is presented with the permutations of a fixed pattern class $\mathcal{X}$ as a set of inputs then the set of all possible outputs is also a pattern class $\mathcal{X}^*$.

We study the map $\mathcal{X} \longrightarrow \mathcal{X}^*$ defined on the set of all pattern classes.

Simple examples of the $\mathcal{X} \longrightarrow \mathcal{X}^*$ map

- Suppose $\mathcal{X}$ is the set of all increasing permutations $12\cdots n$ (one permutation of every length). No priority queue computation can disorder $12\cdots n$ so $\mathcal{X}^* = \mathcal{X}$.
- Suppose $\mathcal{X}$ is the set of all decreasing permutations $n\cdots 21$ (one permutation of every length). Priority queue computations are now just as though the priority queue was a stack. So $\mathcal{X}^*$ is the set of permutations that avoid 132.

Basis of $\mathcal{X}$	Basis of $\mathcal{X}^*$
21 12	21 132

Simple examples of the $\mathcal{X} \longrightarrow \mathcal{X}^*$ map

- Suppose $\mathcal{X}$ is the set of all increasing permutations $12 \cdots n$ (one permutation of every length). No priority queue computation can disorder $12 \cdots n$ so $\mathcal{X}^* = \mathcal{X}$.
- Suppose $\mathcal{X}$ is the set of all decreasing permutations $n \cdots 21$ (one permutation of every length). Priority queue computations are now just as though the priority queue was a stack. So $\mathcal{X}^*$ is the set of permutations that avoid 132.

Basis of $\mathcal{X}$	Basis of $\mathcal{X}^*$
21 12	21 132

Simple examples of the $\mathcal{X} \longrightarrow \mathcal{X}^*$ map

- Suppose $\mathcal{X}$ is the set of all increasing permutations $12\cdots n$ (one permutation of every length). No priority queue computation can disorder $12\cdots n$ so $\mathcal{X}^* = \mathcal{X}$.
- Suppose $\mathcal{X}$ is the set of all decreasing permutations $n\cdots 21$ (one permutation of every length). Priority queue computations are now just as though the priority queue was a stack. So $\mathcal{X}^*$ is the set of permutations that avoid 132.

Basis of $\mathcal{X}$	Basis of $\mathcal{X}^*$
21	21
12	132

The main theorems

Basis of $\mathcal{X}$	Basis of $\mathcal{X}^*$
321	321
312	3142, 4132
231	2431
213	2143
132	1432
123	13254, 14253, 15243

The main theorems continued.

Basis of $\mathcal{X}$	Basis of $\mathcal{X}^*$
132, 321	321, 2143, 2413
213, 321	321, 2143, 2413
231, 312	2413, 2431, 3142, 4132
231, 321	231, 321
123, 132	1423, 1432, 13254
...	...
Any length 3 permutations	Finite basis
...	...

But, if $\mathcal{X} = \mathrm{Av}(2431)$, $\mathcal{X}^*$ is not finitely based.

The main theorems continued.

Basis of $\mathcal{X}$	Basis of $\mathcal{X}^*$
132, 321	321, 2143, 2413
213, 321	321, 2143, 2413
231, 312	2413, 2431, 3142, 4132
231, 321	231, 321
123, 132	1423, 1432, 13254
...	...
Any length 3 permutations	Finite basis
...	...

But, if $\mathcal{X} = \mathrm{Av}(2431)$, $\mathcal{X}^*$ is not finitely based.

When is $\mathcal{X} = \mathcal{X}^*$?

Theorem

$\mathcal{X} = \mathcal{X}^*$ if and only if $\mathcal{X}$ is closed downwards in the weak order.

It is easily detectable from the basis of $\mathcal{X}$ when the weak order condition holds (every basis element β has the property that all permutations above β in the weak order must have some (other) basis element as a subpermutation).

When is $\mathcal{X} = \mathcal{X}^*$?

Theorem

$\mathcal{X} = \mathcal{X}^*$ *if and only if* $\mathcal{X}$ *is closed downwards in the weak order.*

It is easily detectable from the basis of $\mathcal{X}$ when the weak order condition holds (every basis element β has the property that all permutations above β in the weak order must have some (other) basis element as a subpermutation).

When is $\mathcal{X} = \mathcal{X}^*$?

Theorem

$\mathcal{X} = \mathcal{X}^*$ if and only if $\mathcal{X}$ is closed downwards in the weak order.

Corollary

If $\mathcal{X} = \mathrm{Av}(k\ \cdots\ 2\ 1)$ then $\mathcal{X} = \mathcal{X}^*$.

It is easily detectable from the basis of $\mathcal{X}$ when the weak order condition holds (every basis element β has the property that all permutations above β in the weak order must have some (other) basis element as a subpermutation).

Easy proof techniques

Sometimes we know so much about the structure of a pattern class that we can directly compute the outputs of a priority queue given an input permutation.

Example

If $\mathcal{X} = \mathrm{Av}(231, 312)$ then $\mathcal{X}^* = \mathrm{Av}(2413, 2431, 3142, 4132)$

Proof.

Permutations of $\mathcal{X}$ are sums of decreasing permutations. Hence $\mathcal{X}^*$ consists of permutations that are sums of 132-avoiding permutations. □

Easy proof techniques

Sometimes we know so much about the structure of a pattern class that we can directly compute the outputs of a priority queue given an input permutation.

Example

If $\mathcal{X} = \mathrm{Av}(231, 312)$ then $\mathcal{X}^* = \mathrm{Av}(2413, 2431, 3142, 4132)$

Proof.

Permutations of $\mathcal{X}$ are sums of decreasing permutations. Hence $\mathcal{X}^*$ consists of permutations that are sums of 132-avoiding permutations. □

Main proof technique

Given a permutation $\tau = t_1 t_2 \cdots t_n$ we define a partial order $P(\tau)$ on $\{1, \ldots, n\}$ by defining $t_i \prec t_j$ if either

- $i < j$ and $t_i > t_j$, or
- $i < j$ and, for some k with $i < k < j$, $t_i t_k t_j \sim 132$

Example

If $\tau = 31524$ then $P(\tau)$ has constraints $5 \prec 2$, $5 \prec 4$, $\{3, 1\} \prec \{2, 4\}$, and $3 \prec 1$.

Lemma

(σ, τ) is allowable if and only if σ is a linear extension of $P(\tau)$.

Main proof technique

Given a permutation $\tau = t_1 t_2 \cdots t_n$ we define a partial order $P(\tau)$ on $\{1, \ldots, n\}$ by defining $t_i \prec t_j$ if either

- $i < j$ and $t_i > t_j$, or
- $i < j$ and, for some k with $i < k < j$, $t_i t_k t_j \sim 132$

Example

If $\tau = 31524$ then $P(\tau)$ has constraints $5 \prec 2$, $5 \prec 4$, $\{3, 1\} \prec \{2, 4\}$, and $3 \prec 1$.

Lemma

(σ, τ) is allowable if and only if σ is a linear extension of $P(\tau)$.

Main proof technique

Given a permutation $\tau = t_1 t_2 \cdots t_n$ we define a partial order $P(\tau)$ on $\{1, \ldots, n\}$ by defining $t_i \prec t_j$ if either

- $i < j$ and $t_i > t_j$, or
- $i < j$ and, for some k with $i < k < j$, $t_i t_k t_j \sim 132$

Example

If $\tau = 31524$ then $P(\tau)$ has constraints $5 \prec 2$, $5 \prec 4$, $\{3, 1\} \prec \{2, 4\}$, and $3 \prec 1$.

Lemma

(σ, τ) is allowable if and only if σ is a linear extension of $P(\tau)$.

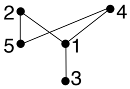
Main proof technique

Given a permutation $\tau = t_1 t_2 \cdots t_n$ we define a partial order $P(\tau)$ on $\{1, \ldots, n\}$ by defining $t_i \prec t_j$ if either

- $i < j$ and $t_i > t_j$, or
- $i < j$ and, for some k with $i < k < j$, $t_i t_k t_j \sim 132$

Example

If $\tau = 31524$ then $P(\tau)$ has constraints $5 \prec 2$, $5 \prec 4$, $\{3, 1\} \prec \{2, 4\}$, and $3 \prec 1$.



Linear extensions
31524 31542 35124
35142 53124 53142

Lemma

(σ, τ) is allowable if and only if σ is a linear extension of $P(\tau)$.

Principal classes

- Suppose $\mathcal{X} = \mathrm{Av}(\alpha)$ and we want the basis for $\mathcal{X}^*$.
- In order that $\tau \notin \mathcal{X}^*$ we require that none of the linear extensions of $P(\tau)$ belong to $\mathcal{X}$.
- So all the linear extensions of $P(\tau)$ must contain α.
- It is sufficient that $P(\tau)$ contains a chain $a_1 \prec a_2 \cdots \prec a_r$ with $a_1 a_2 \cdots a_r \sim \alpha$ (an α-chain).
- So *some* basis elements τ of $\mathcal{X}^*$ may be found by taking $\tau = \cdots a_1 \cdots a_2 \cdots \ldots \cdots a_r \cdots$ where, in between the a_i, we put elements to ensure $a_i \prec a_{i+1}$ in $P(\tau)$.
- If $a_i > a_{i+1}$ then automatically $a_i \prec a_{i+1}$
- If $a_i < a_{i+1}$ then we need $\cdots a_i c_i a_{i+1} \cdots$ with $c_i > a_{i+1}$

Principal classes

- Suppose $\mathcal{X} = \mathrm{Av}(\alpha)$ and we want the basis for $\mathcal{X}^*$.
- In order that $\tau \notin \mathcal{X}^*$ we require that none of the linear extensions of $P(\tau)$ belong to $\mathcal{X}$.
- So all the linear extensions of $P(\tau)$ must contain α.
- It is sufficient that $P(\tau)$ contains a chain $a_1 \prec a_2 \cdots \prec a_r$ with $a_1 a_2 \cdots a_r \sim \alpha$ (an α-chain).
- So *some* basis elements τ of $\mathcal{X}^*$ may be found by taking $\tau = \cdots a_1 \cdots a_2 \cdots \ldots \cdots a_r \cdots$ where, in between the a_i, we put elements to ensure $a_i \prec a_{i+1}$ in $P(\tau)$.
- If $a_i > a_{i+1}$ then automatically $a_i \prec a_{i+1}$
- If $a_i < a_{i+1}$ then we need $\cdots a_i c_i a_{i+1} \cdots$ with $c_i > a_{i+1}$

Principal classes

- Suppose $\mathcal{X} = \mathrm{Av}(\alpha)$ and we want the basis for $\mathcal{X}^*$.
- In order that $\tau \notin \mathcal{X}^*$ we require that none of the linear extensions of $P(\tau)$ belong to $\mathcal{X}$.
- So all the linear extensions of $P(\tau)$ must contain α.
- It is sufficient that $P(\tau)$ contains a chain $a_1 \prec a_2 \cdots \prec a_r$ with $a_1 a_2 \cdots a_r \sim \alpha$ (an α-chain).
- So *some* basis elements τ of $\mathcal{X}^*$ may be found by taking $\tau = \cdots a_1 \cdots a_2 \cdots \ldots \cdots a_r \cdots$ where, in between the a_i, we put elements to ensure $a_i \prec a_{i+1}$ in $P(\tau)$.
- If $a_i > a_{i+1}$ then automatically $a_i \prec a_{i+1}$
- If $a_i < a_{i+1}$ then we need $\cdots a_i c_i a_{i+1} \cdots$ with $c_i > a_{i+1}$

Principal classes

- Suppose $\mathcal{X} = \mathrm{Av}(\alpha)$ and we want the basis for $\mathcal{X}^*$.
- In order that $\tau \notin \mathcal{X}^*$ we require that none of the linear extensions of $P(\tau)$ belong to $\mathcal{X}$.
- So all the linear extensions of $P(\tau)$ must contain α.
- It is sufficient that $P(\tau)$ contains a chain $a_1 \prec a_2 \cdots \prec a_r$ with $a_1 a_2 \cdots a_r \sim \alpha$ (an α-chain).
- So *some* basis elements τ of $\mathcal{X}^*$ may be found by taking $\tau = \cdots a_1 \cdots a_2 \cdots \ldots \cdots a_r \cdots$ where, in between the a_i, we put elements to ensure $a_i \prec a_{i+1}$ in $P(\tau)$.
- If $a_i > a_{i+1}$ then automatically $a_i \prec a_{i+1}$
- If $a_i < a_{i+1}$ then we need $\cdots a_i c_i a_{i+1} \cdots$ with $c_i > a_{i+1}$

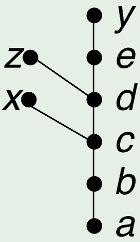
Principal classes

- Suppose $\mathcal{X} = \mathrm{Av}(\alpha)$ and we want the basis for $\mathcal{X}^*$.
- In order that $\tau \notin \mathcal{X}^*$ we require that none of the linear extensions of $P(\tau)$ belong to $\mathcal{X}$.
- So all the linear extensions of $P(\tau)$ must contain α.
- It is sufficient that $P(\tau)$ contains a chain $a_1 \prec a_2 \cdots \prec a_r$ with $a_1 a_2 \cdots a_r \sim \alpha$ (an α-chain).
- So *some* basis elements τ of $\mathcal{X}^*$ may be found by taking $\tau = \cdots a_1 \cdots a_2 \cdots \ldots \cdots a_r \cdots$ where, in between the a_i, we put elements to ensure $a_i \prec a_{i+1}$ in $P(\tau)$.
- If $a_i > a_{i+1}$ then automatically $a_i \prec a_{i+1}$
- If $a_i < a_{i+1}$ then we need $\cdots a_i c_i a_{i+1} \cdots$ with $c_i > a_{i+1}$

Some basis elements

Example

Let $\mathcal{X} = \mathrm{Av}(13524)$. Some elements in the basis of $\mathcal{X}^*$ arise from permutations *axcyebzd* where *acebd* $\sim$ 13524 and $x > c, y > e, z > d$.



There are 15 such permutations.

Sometimes we are lucky

In general this method only generates *some* of the basis elements. But for $\mathrm{Av}(\alpha)$ with $|\alpha| \leq 3$ it produces them all because of:

Theorem

Suppose $|\alpha| \leq 3$*. For all* τ*,* $P(\tau)$ *has no* α*-chain implies* $P(\tau)$ *has an* α*-avoiding linear extension.*

Corollary

Suppose $|\alpha| \leq 3$ *and* $X = \mathrm{Av}(\alpha)$*. The basis elements of* $\mathcal{X}^*$ *are those minimal* τ *for which* $P(\tau)$ *has an* α*-chain.*

Proof.

6 different cases! □

One of the easy cases: $\alpha = 312$

- Consider any $\tau = m_1\tau_1 m_2\tau_2 \cdots m_k\tau_k$ with left to right maxima m_i and where $P(\tau)$ has no 312-chain
- Define $\lambda = m_1\lambda_1 m_2\lambda_2 \cdots m_k\lambda_k$ with λ_i decreasing.

1. λ is a linear extension of $P(\tau)$

   The only problem could be that, in some τ_i, we have $s \prec t$ with $s < t$ (which would prevent our arranging λ_i in decreasing order). But then $m_i \prec s \prec t$ would be a 312-chain in $P(\tau)$.

2. λ does not contain 312

   If λ contains some 312 we can take one of the form $m_i ab$ with $a \in \lambda_i$ and $b \in \lambda_j$ for some $j > i$. But, because of $am_jb \sim 132$ we have $a \prec b$. In addition $m_i \prec a$ because $m_i > a$ and so $m_i \prec a \prec b$ is a 312-chain in $P(\tau)$.

One of the easy cases: $\alpha = 312$

- Consider any $\tau = m_1\tau_1 m_2\tau_2 \cdots m_k\tau_k$ with left to right maxima m_i and where $P(\tau)$ has no 312-chain
- Define $\lambda = m_1\lambda_1 m_2\lambda_2 \cdots m_k\lambda_k$ with λ_i decreasing.

1. λ is a linear extension of $P(\tau)$

   The only problem could be that, in some τ_i, we have $s \prec t$ with $s < t$ (which would prevent our arranging λ_i in decreasing order). But then $m_i \prec s \prec t$ would be a 312-chain in $P(\tau)$.

2. λ does not contain 312

   If λ contains some 312 we can take one of the form $m_i a b$ with $a \in \lambda_i$ and $b \in \lambda_j$ for some $j > i$. But, because of $a m_i b \sim 132$ we have $a \prec b$. In addition $m_i \prec a$ because $m_i > a$ and so $m_i \prec a \prec b$ is a 312-chain in $P(\tau)$.

One of the easy cases: $\alpha = 312$

- Consider any $\tau = m_1\tau_1 m_2\tau_2 \cdots m_k\tau_k$ with left to right maxima m_i and where $P(\tau)$ has no 312-chain
- Define $\lambda = m_1\lambda_1 m_2\lambda_2 \cdots m_k\lambda_k$ with λ_i decreasing.

1. λ is a linear extension of $P(\tau)$

   The only problem could be that, in some τ_i, we have $s \prec t$ with $s < t$ (which would prevent our arranging λ_i in decreasing order). But then $m_i \prec s \prec t$ would be a 312-chain in $P(\tau)$.

2. λ does not contain 312

   If λ contains some 312 we can take one of the form $m_i ab$ with $a \in \lambda_i$ and $b \in \lambda_j$ for some $j > i$. But, because of $am_jb \sim 132$ we have $a \prec b$. In addition $m_i \prec a$ because $m_i > a$ and so $m_i \prec a \prec b$ is a 312-chain in $P(\tau)$.

One of the easy cases: $\alpha = 312$

- Consider any $\tau = m_1\tau_1 m_2\tau_2 \cdots m_k\tau_k$ with left to right maxima m_i and where $P(\tau)$ has no 312-chain
- Define $\lambda = m_1\lambda_1 m_2\lambda_2 \cdots m_k\lambda_k$ with λ_i decreasing.

1. λ is a linear extension of $P(\tau)$

   The only problem could be that, in some τ_i, we have $s \prec t$ with $s < t$ (which would prevent our arranging λ_i in decreasing order). But then $m_i \prec s \prec t$ would be a 312-chain in $P(\tau)$.

2. λ does not contain 312

   If λ contains some 312 we can take one of the form $m_i ab$ with $a \in \lambda_i$ and $b \in \lambda_j$ for some $j > i$. But, because of $am_jb \sim 132$ we have $a \prec b$. In addition $m_i \prec a$ because $m_i > a$ and so $m_i \prec a \prec b$ is a 312-chain in $P(\tau)$.

One of the easy cases: $\alpha = 312$

- Consider any $\tau = m_1\tau_1 m_2\tau_2 \cdots m_k\tau_k$ with left to right maxima m_i and where $P(\tau)$ has no 312-chain
- Define $\lambda = m_1\lambda_1 m_2\lambda_2 \cdots m_k\lambda_k$ with λ_i decreasing.

1. λ is a linear extension of $P(\tau)$

   The only problem could be that, in some τ_i, we have $s \prec t$ with $s < t$ (which would prevent our arranging λ_i in decreasing order). But then $m_i \prec s \prec t$ would be a 312-chain in $P(\tau)$.

2. λ does not contain 312

   If λ contains some 312 we can take one of the form $m_i ab$ with $a \in \lambda_i$ and $b \in \lambda_j$ for some $j > i$. But, because of $am_jb \sim 132$ we have $a \prec b$. In addition $m_i \prec a$ because $m_i > a$ and so $m_i \prec a \prec b$ is a 312-chain in $P(\tau)$.

One of the easy cases: $\alpha = 312$

- Consider any $\tau = m_1\tau_1 m_2\tau_2 \cdots m_k\tau_k$ with left to right maxima m_i and where $P(\tau)$ has no 312-chain
- Define $\lambda = m_1\lambda_1 m_2\lambda_2 \cdots m_k\lambda_k$ with λ_i decreasing.

1. λ is a linear extension of $P(\tau)$

   The only problem could be that, in some τ_i, we have $s \prec t$ with $s < t$ (which would prevent our arranging λ_i in decreasing order). But then $m_i \prec s \prec t$ would be a 312-chain in $P(\tau)$.

2. λ does not contain 312

   If λ contains some 312 we can take one of the form $m_i ab$ with $a \in \lambda_i$ and $b \in \lambda_j$ for some $j > i$. But, because of $am_jb \sim 132$ we have $a \prec b$. In addition $m_i \prec a$ because $m_i > a$ and so $m_i \prec a \prec b$ is a 312-chain in $P(\tau)$.

Further work and open problems

- Characterize the principal classes $\mathcal{X} = \mathrm{Av}(\alpha)$ for which $\mathcal{X}^*$ is finitely based. We have some suggestive numerical results for all α with $|\alpha| = 4$.
- Solve the 'opposite' problem: given a pattern class $\mathcal{X}$, what is the pattern class of *inputs* that gives rise to $\mathcal{X}$ as a set of *outputs*. This may be an easier problem since, if $\mathcal{X}$ contains every increasing permutation, the answer is "All permutations".
- Carry out similar investigations for pair pattern classes other than $\mathcal{A}$. This almost certainly rather hard since pair pattern classes are far less studied than ordinary pattern classes.

— That's all folks —

Further work and open problems

- Characterize the principal classes $\mathcal{X} = \mathrm{Av}(\alpha)$ for which $\mathcal{X}^*$ is finitely based. We have some suggestive numerical results for all α with $|\alpha| = 4$.
- Solve the 'opposite' problem: given a pattern class $\mathcal{X}$, what is the pattern class of *inputs* that gives rise to $\mathcal{X}$ as a set of *outputs*. This may be an easier problem since, if $\mathcal{X}$ contains every increasing permutation, the answer is "All permutations".
- Carry out similar investigations for pair pattern classes other than $\mathcal{A}$. This almost certainly rather hard since pair pattern classes are far less studied than ordinary pattern classes.

— That's all folks —

Further work and open problems

- Characterize the principal classes $\mathcal{X} = \mathrm{Av}(\alpha)$ for which $\mathcal{X}^*$ is finitely based. We have some suggestive numerical results for all α with $|\alpha| = 4$.
- Solve the 'opposite' problem: given a pattern class $\mathcal{X}$, what is the pattern class of *inputs* that gives rise to $\mathcal{X}$ as a set of *outputs*. This may be an easier problem since, if $\mathcal{X}$ contains every increasing permutation, the answer is "All permutations".
- Carry out similar investigations for pair pattern classes other than $\mathcal{A}$. This almost certainly rather hard since pair pattern classes are far less studied than ordinary pattern classes.

— That's all folks —

Further work and open problems

- Characterize the principal classes $\mathcal{X} = \mathrm{Av}(\alpha)$ for which $\mathcal{X}^*$ is finitely based. We have some suggestive numerical results for all α with $|\alpha| = 4$.
- Solve the 'opposite' problem: given a pattern class $\mathcal{X}$, what is the pattern class of *inputs* that gives rise to $\mathcal{X}$ as a set of *outputs*. This may be an easier problem since, if $\mathcal{X}$ contains every increasing permutation, the answer is "All permutations".
- Carry out similar investigations for pair pattern classes other than $\mathcal{A}$. This almost certainly rather hard since pair pattern classes are far less studied than ordinary pattern classes.

— That's all folks —