Antichains and the Structure of Permutation Classes

Robert Brignall

Heilbronn Institute for Mathematical Research
University of Bristol

Thursday 13th May, 2010
Outline

1 Introduction
   • Permutation classes
   • Enumeration
   • Partial well-order and antichains

2 Simple permutations
   • Intervals
   • Substitution decomposition
   • Finitely many simples

3 Grid classes
   • Introduction
   • Monotone classes and partial well-order
   • Far beyond monotone
   • Nearly monotone

4 Summary
Outline

1 Introduction
- Permutation classes
- Enumeration
- Partial well-order and antichains

2 Simple permutations
- Intervals
- Substitution decomposition
- Finitely many simples

3 Grid classes
- Introduction
- Monotone classes and partial well-order
- Far beyond monotone
- Nearly monotone

4 Summary
Setting the Scene

- **Permutation** of length $n$: an ordering on the symbols $1, \ldots, n$.
- For example: $\pi = 15482763$. 
Permutation of length $n$: an ordering on the symbols $1, \ldots, n$.

For example: $\pi = 15482763$.

Graphical viewpoint: plot the points $(i, \pi(i))$. 

Example
Knuth (1969): what permutations can be sorted through a stack?
Knuth (1969): what permutations can be sorted through a stack?
Knuth (1969): what permutations can be sorted through a stack?

Example
Knuth (1969): what permutations can be sorted through a stack?

Example

```
13
2
4
```
Knuth (1969): what permutations can be sorted through a stack?
Knuth (1969): what permutations can be sorted through a stack?
Knuth (1969): what permutations can be sorted through a stack?

Example

```
12
  |
  v
  4
  |
  \---\---\---
    3
```

13th May 2010
Knuth (1969): what permutations can be sorted through a stack?

Example
Knuth (1969): what permutations can be sorted through a stack?
Knuth (1969): what permutations can be sorted through a stack?

Example

1234
Knuth (1969): what permutations can be sorted through a stack?
Knuth (1969): what permutations can be sorted through a stack?

Example
Knuth (1969): what permutations can be sorted through a stack?

Example

![Diagram of a stack sorting example with numbers 1, 2, and 3]
Knuth (1969): what permutations can be sorted through a stack?

Example

231 is not stack-sortable.
Stack Sorting

- **Knuth (1969):** what permutations can be sorted through a stack?

**Example**

- 231 is not stack-sortable.
- In general: can’t sort any permutation with a subsequence $abc$ such that $c < a < b$. ($abc$ forms a 231 “pattern”.)

Containment

A permutation $\tau = \tau(1) \cdots \tau(k)$ is contained in the permutation $\sigma = \sigma(1)\sigma(2) \cdots \sigma(n)$ if there exists a subsequence $\sigma(i_1)\sigma(i_2) \cdots \sigma(i_k)$ order isomorphic to $\tau$. 
A permutation $\tau = \tau(1) \cdots \tau(k)$ is contained in the permutation $\sigma = \sigma(1)\sigma(2)\cdots\sigma(n)$ if there exists a subsequence $\sigma(i_1)\sigma(i_2)\cdots\sigma(i_k)$ order isomorphic to $\tau$. 

Example

\[1 \quad 3 \quad 5 \quad 2 \quad 4\]

\[4 \quad 2 \quad 1 \quad 6 \quad 3 \quad 8 \quad 5 \quad 7\]
A permutation $\tau = \tau(1) \cdots \tau(k)$ is contained in the permutation $\sigma = \sigma(1)\sigma(2) \cdots \sigma(n)$ if there exists a subsequence $\sigma(i_1)\sigma(i_2) \cdots \sigma(i_k)$ order isomorphic to $\tau$.

**Example**

![Example diagram](image-url)
Containment forms a partial order on the set of all permutations. (Reflexive, antisymmetric, transitive.)

Downwards-closed sets in this partial order form permutation classes. i.e. $\pi \in \mathcal{C}$ and $\sigma \leq \pi$ implies $\sigma \in \mathcal{C}$. 
Permutation Classes

- Containment forms a **partial order** on the set of all permutations. (Reflexive, antisymmetric, transitive.)

- Downwards-closed sets in this partial order form permutation classes. i.e. $\pi \in \mathcal{C}$ and $\sigma \leq \pi$ implies $\sigma \in \mathcal{C}$.

- A permutation class $\mathcal{C}$ can be seen to avoid certain permutations. Write $\mathcal{C} = \text{Av}(B) = \{\pi : \beta \nleq \pi \text{ for all } \beta \in B\}$.

- The minimal avoidance set is the **basis**. It is **unique** but need not be **finite**.

- E.g. the stack-sortable permutations are $\text{Av}(231)$. 
Permutation Classes

- Containment forms a **partial order** on the set of all permutations.
  (Reflexive, antisymmetric, transitive.)
- Downwards-closed sets in this partial order form permutation classes.
  i.e. $\pi \in \mathcal{C}$ and $\sigma \leq \pi$ implies $\sigma \in \mathcal{C}$.
- A permutation class $\mathcal{C}$ can be seen to avoid certain permutations.
  Write $\mathcal{C} = \text{Av}(B) = \{ \pi : \beta \nleq \pi \text{ for all } \beta \in B \}$.
- The minimal avoidance set is the basis. It is unique but need not be finite.
- E.g. the stack-sortable permutations are $\text{Av}(231)$.
- Graph theoretic analogue: **hereditary properties of graphs**
  (e.g. triangle-free graphs, planar graphs, ...).
Easy Examples

- $\text{Av}(21) = \{1, 12, 123, 1234, \ldots\}$, the increasing permutations.
- $\text{Av}(12) = \{1, 21, 321, 4321, \ldots\}$, the decreasing permutations.
Easy Examples

\[ \oplus 21 = \text{Av}(321, 312, 231) = \{1, 12, 21, 123, 132, 213, \ldots \}. \]
\[ \ominus 12 = \text{Av}(123, 213, 132) = \{1, 12, 21, 231, 312, 321, \ldots \}. \]
Exact Enumeration

- $C_n$ – permutations in $C$ of length $n$.
- $\sum |C_n| x^n$ is the generating function.

Example

The generating function of $C = \text{Av}(12)$ is:

$$1 + x + x^2 + x^3 + \cdots = \frac{1}{1 - x}$$
Asymptotic Enumeration

Theorem (Marcus and Tardos, 2004)

For every permutation class $C$ other than the class of all permutations, there exists a constant $K$ such that

$$\limsup_{n \to \infty} \sqrt[n]{|C_n|} \leq K.$$

- Upper growth rate of $C$ is $\limsup_{n \to \infty} \sqrt[n]{|C_n|}$. 
Asymptotic Enumeration

Theorem (Marcus and Tardos, 2004)

For every permutation class $C$ other than the class of all permutations, there exists a constant $K$ such that

$$\limsup_{n \to \infty} \sqrt[n]{|C_n|} \leq K.$$ 

- Upper growth rate of $C$ is $\limsup_{n \to \infty} \sqrt[n]{|C_n|}$.

- Big open question: does the growth rate $\lim_{n \to \infty} \sqrt[n]{|C_n|}$, always exist?
Av(321) vs Av(231)

- Stack sortable permutations Av(231) enumerated by the Catalan numbers. Generating function:

\[ f(x) = \frac{1 - \sqrt{1 - 4x}}{2x} = 1 + x + 2x^2 + 5x^3 + 14x^4 + \ldots \]
Av(321) vs Av(231)

- Stack sortable permutations Av(231) enumerated by the Catalan numbers. Generating function:

  \[ f(x) = \frac{1 - \sqrt{1 - 4x}}{2x} = 1 + x + 2x^2 + 5x^3 + 14x^4 + \ldots \]

- Using the Robinson-Schensted-Knuth correspondence with Young Tableaux, \(|\text{Av}(321)|_n = |\text{Av}(231)|_n\).
Av(321) vs Av(231)

- Stack sortable permutations Av(231) enumerated by the Catalan numbers. Generating function:

\[ f(x) = \frac{1 - \sqrt{1 - 4x}}{2x} = 1 + x + 2x^2 + 5x^3 + 14x^4 + \ldots \]

- Using the Robinson-Schensted-Knuth correspondence with Young Tableaux, \(|\text{Av}(321)|_n = |\text{Av}(231)|_n\).

- Despite being equinumerous, these two classes are very different: Av(321) contains infinite antichains and hence has uncountably many subclasses, while Av(231) does not.
(Infinite) set of pairwise incomparable permutations.
(Infinite) set of pairwise incomparable permutations.

N.B. These permutations avoid 321.
(Infinite) set of pairwise incomparable permutations.

Example (Increasing Oscillating Antichain)

Bottom copies of 4123 must match up: the anchor.
Infinite Antichains

- (Infinite) set of pairwise incomparable permutations.

Example (Increasing Oscillating Antichain)

- Each point is matched in turn.
(Infinite) set of pairwise incomparable permutations.

Example (Increasing Oscillating Antichain)

Each point is matched in turn.
(Infinite) set of pairwise incomparable permutations.

Each point is matched in turn.
Infinite Antichains

- (Infinite) set of pairwise incomparable permutations.

Example (Increasing Oscillating Antichain)

- Last pair cannot be embedded.
When are there antichains?

No infinite antichains.
- Words over a finite alphabet [Higman].
- Graphs closed under minors [Robertson and Seymour].

Infinite antichains.
- Graphs closed under induced subgraphs (or merely subgraphs). e.g. $C_3, C_4, C_5, \ldots$
- Permutations closed under containment.
- Tournaments, digraphs, \ldots
A permutation class is \textit{partially well-ordered} (pwo) if it contains no infinite antichains.
Partial Well Order

A permutation class is partially well-ordered (pwo) if it contains no infinite antichains.

**Question**

*Can we decide whether a permutation class given by a finite basis is pwo?*

- To prove pwo — **Higman’s theorem** is useful.
- To prove not pwo — find an antichain.
Partial Well Order

- A permutation class is **partially well-ordered** (pwo) if it contains no infinite antichains.

**Question**

*Can we decide whether a hereditary property given by a finite basis is wqo?*

- To prove pwo — **Higman’s theorem** is useful.
- To prove not pwo — find an antichain.
- Other structures: **well quasi-order**, not pwo, but same idea.
Outline

1 Introduction
   - Permutation classes
   - Enumeration
   - Partial well-order and antichains

2 Simple permutations
   - Intervals
   - Substitution decomposition
   - Finitely many simples

3 Grid classes
   - Introduction
   - Monotone classes and partial well-order
   - Far beyond monotone
   - Nearly monotone

4 Summary
Intervals

- Pick any permutation $\pi$.
- An interval of $\pi$ is a set of contiguous indices $I = [a, b]$ such that $\pi(I) = \{\pi(i) : i \in I\}$ is also contiguous.
Pick any permutation $\pi$.

An interval of $\pi$ is a set of contiguous indices $I = [a, b]$ such that $\pi(I) = \{\pi(i) : i \in I\}$ is also contiguous.
Pick any permutation $\pi$.

An interval of $\pi$ is a set of contiguous indices $l = [a, b]$ such that $\pi(l) = \{\pi(i) : i \in l\}$ is also contiguous.

Example
Pick any permutation $\pi$.

An interval of $\pi$ is a set of contiguous indices $I = [a, b]$ such that $\pi(I) = \{\pi(i) : i \in I\}$ is also contiguous.

Intervals are important in biomathematics (genetic algorithms, matching gene sequences).
Simple Permutations

- A simple permutation: The only intervals are singletons and the whole thing.
A simple permutation: The only intervals are *singletons* and the whole thing.
Simple Permutations

- A simple permutation: The only intervals are singletons and the whole thing.
A simple permutation: The only intervals are singletons and the whole thing.
A simple permutation: The only intervals are singletons and the whole thing.
A simple permutation: The only intervals are singletons and the whole thing.
A simple permutation: The only intervals are singletons and the whole thing.
Simple Permutations

- A simple permutation: The only intervals are singletons and the whole thing.

Example

- 1 is simple, as are 12 and 21.
- There are no simple permutations of length three.
- Two of length four: 2413 and 3142.
Simple permutations are the “building blocks” of all permutations.
Decomposing Permutations

- Simple permutations are the “building blocks” of all permutations.
- Break permutation into maximal proper intervals.
Simple permutations are the “building blocks” of all permutations.
Break permutation into maximal proper intervals.
Gives a unique simple permutation, the skeleton.
Simple permutations are the “building blocks” of all permutations. If simple has \( > 2 \) points then the blocks are unique.
Decomposing Permutations

- Simple permutations are the “building blocks” of all permutations.
- If simple has $> 2$ points then the blocks are unique.
- This decomposition is the substitution decomposition.

Example

![Diagram of permutation classes](image-url)
Non-uniqueness

- Simple permutation of length 2: block decomposition is not unique.

Example
Non-uniqueness

- Simple permutation of length 2: block decomposition is not unique.
Non-uniqueness

- Simple permutation of length 2: block decomposition is not unique.
Non-uniqueness

- Underlying structure is an increasing permutation.

Example
Finitely Many Simples

Using the substitution decomposition, we can say a lot about permutation classes that contain only finitely many simples [Albert and Atkinson, 2005]:
Finitely Many Simples

Using the substitution decomposition, we can say a lot about permutation classes that contain only finitely many simples [Albert and Atkinson, 2005]:

- They have a finite basis.
- They are enumerated by algebraic generating functions.
- They are partially well-ordered.
Finitely Many Simples

Using the substitution decomposition, we can say a lot about permutation classes that contain only finitely many simples [Albert and Atkinson, 2005]:

- They have a finite basis.
- They are enumerated by algebraic generating functions.
- They are partially well-ordered.

**Theorem (B., Ruškuc and Vatter, 2008)**

*It is possible to decide whether a permutation class given by a finite basis contains infinitely many simple permutations.*
Finitely Many Simples

Using the substitution decomposition, we can say a lot about permutation classes that contain only finitely many simples [Albert and Atkinson, 2005]:

- They have a finite basis.
- They are enumerated by algebraic generating functions.
- They are partially well-ordered.

**Theorem (B., Ruškuc and Vatter, 2008)**

*It is possible to decide whether a permutation class given by a finite basis contains infinitely many simple permutations.*

- There should be a graph-theoretic analogue of this result!
Finitely Many Simples $\Rightarrow$ Partially Well-Ordered

- Take a class $\mathcal{C}$ containing a finite set $S$ of simple permutations.
- Every permutation in $\mathcal{C}$ has a skeleton from $S$. 
Finitely Many Simples $\Rightarrow$ Partially Well-Ordered

- Take a class $\mathcal{C}$ containing a finite set $S$ of simple permutations.
- Every permutation in $\mathcal{C}$ has a skeleton from $S$.
- Think of each $\sigma \in S$ of length $n$ as an $n$-ary operation.
- Starting with the permutation $1$, we build every permutation in the class $\mathcal{C}$ by recursively using this finite set of operations.
Finitely Many Simples $\Rightarrow$ Partially Well-Ordered

- Take a class $C$ containing a finite set $S$ of simple permutations.
- Every permutation in $C$ has a skeleton from $S$.
- Think of each $\sigma \in S$ of length $n$ as an $n$-ary operation.
- Starting with the permutation $1$, we build every permutation in the class $C$ by recursively using this finite set of operations.
- Now use Higman's Theorem.
Outline

1. Introduction
   - Permutation classes
   - Enumeration
   - Partial well-order and antichains

2. Simple permutations
   - Intervals
   - Substitution decomposition
   - Finitely many simples

3. Grid classes
   - Introduction
   - Monotone classes and partial well-order
   - Far beyond monotone
   - Nearly monotone

4. Summary
Grid Classes

- **Matrix** $\mathcal{M}$ whose entries are permutation classes.
- **$\text{Grid}(\mathcal{M})$** the grid class of $\mathcal{M}$: all permutations which can be “gridded” so each cell satisfies constraints of $\mathcal{M}$.

**Example**

Let $\mathcal{M} = \begin{pmatrix} \text{Av}(21) & \text{Av}(231) & \emptyset \\ \text{Av}(123) & \emptyset & \text{Av}(12) \end{pmatrix}$.

$\in \text{Grid}(\mathcal{M})$
Grid classes are useful

- Recall: Growth rate of $\mathcal{C}$ is $\lim_{n \to \infty} \sqrt[n]{|\mathcal{C}_n|}$ (if it exists).
Grid classes are useful

- Recall: Growth rate of \( \mathcal{C} \) is \( \lim_{n \to \infty} \sqrt[n]{|\mathcal{C}_n|} \) (if it exists).

- Using grid classes: Below \( \kappa \approx 2.20557 \), growth rates exist and can be characterised [Kaiser and Klazar; Vatter]:

\[
\begin{array}{cccccc}
0 & 1 & \phi & 2 & \kappa \\
\hline
\end{array}
\]

- \( \kappa \) is the lowest growth rate where we encounter infinite antichains, and hence uncountably many permutation classes.
Grid classes are useful

- Recall: Growth rate of $\mathcal{C}$ is $\lim_{n \to \infty} \sqrt[n]{|\mathcal{C}_n|}$ (if it exists).
- Using grid classes: Below $\kappa \approx 2.20557$, growth rates exist and can be characterised [Kaiser and Klazar; Vatter]:

$$
\begin{array}{cccccc}
0 & 1 & \phi & 2 & \kappa \\
\hline
\end{array}
$$

- $\kappa$ is the lowest growth rate where we encounter infinite antichains, and hence uncountably many permutation classes.
- Cf “canonical properties” of graphs [Balogh, Bollobás and Weinreich].
Monotone Grid Classes

- **Special case:** all cells of $\mathcal{M}$ are $\text{Av}(21)$ or $\text{Av}(12)$.
- **Rewrite $\mathcal{M}$** as a matrix with entries in $\{0, 1, -1\}$.

**Example**

$$
\mathcal{M} = \begin{pmatrix}
1 & 1 & 0 \\
-1 & 0 & 1 \\
0 & 1 & -1
\end{pmatrix}
$$
Monotone Grid Classes

- **Special case**: all cells of $\mathcal{M}$ are $\text{Av}(21)$ or $\text{Av}(12)$.
- Rewrite $\mathcal{M}$ as a matrix with entries in $\{0, 1, -1\}$.

**Example**

$$
\mathcal{M} = \begin{pmatrix}
1 & 1 & 0 \\
-1 & 0 & 1 \\
0 & 1 & -1
\end{pmatrix}
$$
The Graph of a Matrix

- **Graph of a matrix**, $G(M)$, formed by connecting together all non-zero entries that share a row or column and are not “separated” by any other nonzero entry.

### Example

\[
\begin{pmatrix}
C & 0 & 0 & D \\
0 & 0 & \mathcal{E} & 0 \\
D & \mathcal{E} & 0 & C \\
0 & 0 & 0 & D
\end{pmatrix}
\]
The Graph of a Matrix

- **Graph of a matrix**, $G(M)$, formed by connecting together all non-zero entries that share a row or column and are not “separated” by any other nonzero entry.

**Example**

$$
\begin{pmatrix}
C & \varepsilon \\
\varepsilon & D
\end{pmatrix}
$$
The Graph of a Matrix

- **Graph of a matrix**, $G(M)$, formed by connecting together all non-zero entries that share a row or column and are not “separated” by any other nonzero entry.

**Example**

$$
\begin{pmatrix}
C & D \\
D & E \\
E & C \\
D & D
\end{pmatrix}
$$
Theorem (Murphy and Vatter, 2003)

The monotone grid class $\text{Grid}(\mathcal{M})$ is pwo if and only if $G(\mathcal{M})$ is a forest, i.e. $G(\mathcal{M})$ contains no cycles.
Theorem (Murphy and Vatter, 2003)

The monotone grid class Grid({\mathcal{M}}) is pwo if and only if G({\mathcal{M}}) is a forest, i.e. G({\mathcal{M}}) contains no cycles.

Proof.

(⇒) Construct infinite antichains that “walk” around a cycle.
When does that apply?

**Question**

*When is a class $C$ (a subset of) a monotone grid class?*
When does that apply?

**Question**

When is a class $C$ (a subset of) a monotone grid class?

**Answer [Huczynska and Vatter]**

A class $C$ is monotone griddable if and only if it contains neither the classes \( \ominus 21 \) nor \( \ominus 12 \).
Non-monotone cells

- If a class is not monotone griddable, then perhaps it can be gridded by a matrix which is mostly monotone:

Example

$$
\begin{pmatrix}
\text{Av}(21) & 0 & 0 & \text{Av}(21) \\
0 & \ominus 12 & 0 & 0 \\
\ominus 21 & 0 & \text{Av}(12) & 0 \\
0 & 0 & 0 & \ominus 21
\end{pmatrix}
$$
Non-monotone cells

- If a class is not monotone griddable, then perhaps it can be gridded by a matrix which is mostly monotone:

Example

\[
\begin{pmatrix}
\text{Av}(21) & \text{Av}(21) \\
\oplus 21 & \text{Av}(12) \\
\end{pmatrix}
\]

- To be pwo, graph must still be a forest, but now the number of non-mono-tone-griddable cells in each component matters.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

WLOG graph is a path connecting two bad cells.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
Two is too many

**Theorem**

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

**Proof.**

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
Two is too many

**Theorem**

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

**Proof.**

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
**Theorem**

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

**Proof.**

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

**Theorem**

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

**Proof.**

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

**Theorem**

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

**Proof.**

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
- Flip and permute back.
Two is too many

**Theorem**

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

**Proof.**

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
- Flip and permute back.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
- Flip and permute back.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
- Flip and permute back.
Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
- Flip and **permute** back.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
- Flip and permute back.
Two is too many

**Theorem**

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

**Proof.**

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
- Flip and **permute** back.
Two is too many

Theorem

A grid class whose graph has a component containing two or more non-monotone-griddable classes is not pwo.

Proof.

- WLOG graph is a path connecting two bad cells.
- Permute rows and columns.
- Flip rows and columns.
- Build antichain with grid pin sequences.
- Flip and permute back.
- Still have an antichain.
Just one non-monotone

- Suppose the bad cell contains only finitely many simple permutations.
Just one non-monotone

- Suppose the bad cell contains only finitely many simple permutations.
- Build permutations component-wise: use the substitution decomposition on the red cell, and view each component as a tree rooted on this cell.
Suppose the bad cell contains only finitely many simple permutations.

Build permutations component-wise: use the substitution decomposition on the red cell, and view each component as a tree rooted on this cell.

This defines a construction for all permutations in the grid class, which is amenable to Higman’s Theorem.
Just one non-monotone

**Theorem**

Let $\mathcal{M}$ be a gridding matrix for which each component is a forest and contains at most one non-monotone cell. If every non-monotone cell contains only finitely many simple permutations, then $\text{Grid}(\mathcal{M})$ is pwo.
But sometimes one is too much...

- One cell containing arbitrarily long increasing oscillations next to a monotone cell is bad...
Outline

1 Introduction
   • Permutation classes
   • Enumeration
   • Partial well-order and antichains

2 Simple permutations
   • Intervals
   • Substitution decomposition
   • Finitely many simples

3 Grid classes
   • Introduction
   • Monotone classes and partial well-order
   • Far beyond monotone
   • Nearly monotone

4 Summary
Two non-monotone per component: class not pwo.

One non-monotone but finitely many simples: class is pwo.
Summary

- **Two** non-monotone per component: class *not pwo*.
- **One** non-monotone but finitely many simples: class is *pwo*.
- **To-do**: one non-monotone but infinitely many simples (some antichains known).
Summary

- Two non-monotone per component: class not pwo.
- One non-monotone but finitely many simples: class is pwo.
- To-do: one non-monotone but infinitely many simples (some antichains known).

Question

Can we decide whether a permutation class given by a finite basis is pwo?

- There are still a lot of obstacles, but maybe we’re a bit closer.
Thanks!