Algorithmic Coincidence Classification of Mesh Patterns

Permutation Patterns 2016

Bjarki Ágúst Guðmundsson
Tómas Ken Magnússon
Henning Ulfarsson

Reykjavík University, Iceland

June 2016

Coincidence

- For classical patterns p, p' it is natural to ask whether they are *Wilf-equivalent*, i.e., $|\mathrm{Av_n}(p)| = |\mathrm{Av_n}(p')|$ for all n

Coincidence

- For classical patterns p, p' it is natural to ask whether they are *Wilf-equivalent*, i.e., $|\mathrm{Av_n}(p)| = |\mathrm{Av_n}(p')|$ for all n
- If we have mesh patterns m, m' we can ask a more basic question: does $\mathrm{Av_n}(m) = \mathrm{Av_n}(m')$ for all n?

Coincidence

- For classical patterns p, p' it is natural to ask whether they are *Wilf-equivalent*, i.e., $|\mathrm{Av_n}(p)| = |\mathrm{Av_n}(p')|$ for all n
- If we have mesh patterns m, m' we can ask a more basic question: does $\mathrm{Av_n}(m) = \mathrm{Av_n}(m')$ for all n?
- When this occurs we say that m and m' are *coincident*, denoted $m \asymp m'$

Coincidence

- For classical patterns p, p' it is natural to ask whether they are *Wilf-equivalent*, i.e., $|\mathrm{Av_n}(p)| = |\mathrm{Av_n}(p')|$ for all n
- If we have mesh patterns m, m' we can ask a more basic question: does $\mathrm{Av_n}(m) = \mathrm{Av_n}(m')$ for all n?

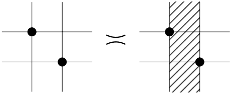
- When this occurs we say that m and m' are *coincident*, denoted $m \asymp m'$
- Note that for classical patterns $p \asymp p'$ if and only if $p = p'$

Example

- A permutation has an inversion if and only if it has a descent

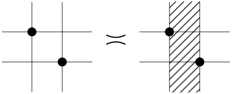
Example

- A permutation has an inversion if and only if it has a descent
- This is the coincidence



Example

- A permutation has an inversion if and only if it has a descent
- This is the coincidence

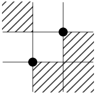
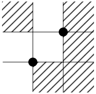


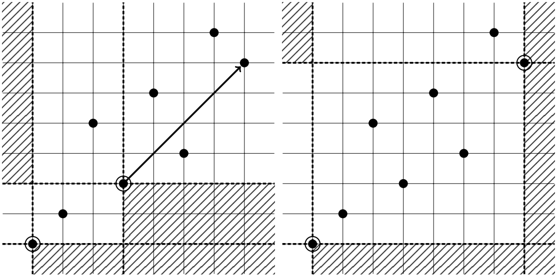
- Mesh patterns are not coincident if they have different underlying classical patterns

Previous work

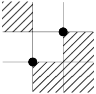
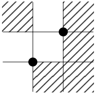
- The shading lemma [2], and the simultaneous shading lemma [1] give sufficient conditions that imply coincidence of patterns
- They are close to capturing all coincidences of mesh patterns of length $\leqslant 2$
- The idea behind the lemmas is swapping out points in an occurrence of a pattern, obtaining the objective pattern

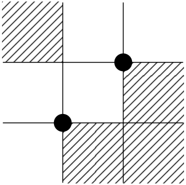
Shading lemma: Example

$p =$  $\asymp$  $= p'$

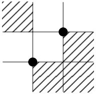
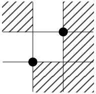


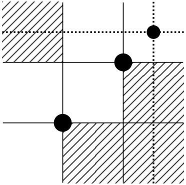
Same example: Different argument

$p =$  $\asymp$  $= p'$

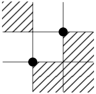
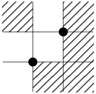


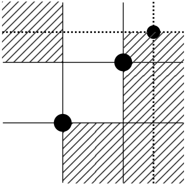
Same example: Different argument

$p =$  $\asymp$  $= p'$

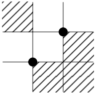
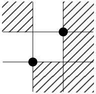


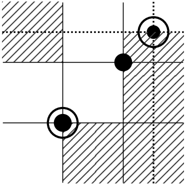
Same example: Different argument

$p =$  $\asymp$  $= p'$

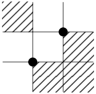
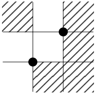


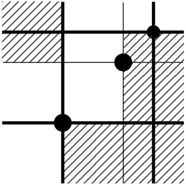
Same example: Different argument

$p =$  $\asymp$  $= p'$

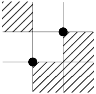
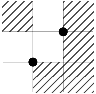


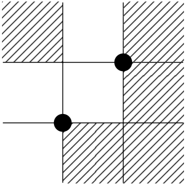
Same example: Different argument

$p =$  $\asymp$  $= p'$



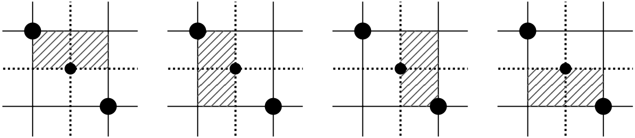
Same example: Different argument

$p =$  $\asymp$  $= p'$



Formalization

Let $p = 21$. The set $p^{\ulcorner 1,1 \urcorner \star}$ consists of the mesh patterns:



$p^{\ulcorner 1,1 \urcorner \uparrow}$ $\quad$ $p^{\overleftarrow{\ulcorner 1,1 \urcorner}}$ $\quad$ $p^{\overrightarrow{\ulcorner 1,1 \urcorner}}$ $\quad$ $p^{\ulcorner 1,1 \urcorner \downarrow}$

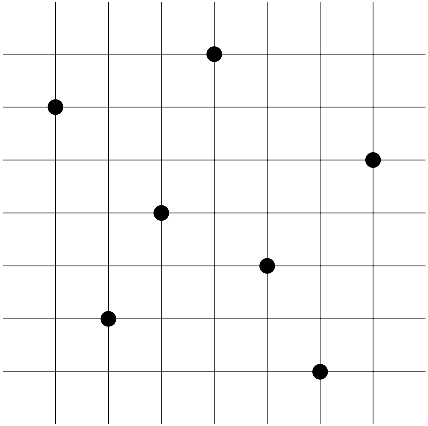
Lemma 2

Let $p = (\tau, R)$ be a mesh pattern with $\ulcorner i, j \urcorner \notin R$. If any mesh pattern in the set $p^{\ulcorner i,j \urcorner \star}$ contains a non-trivial occurrence of p then $p \asymp (\tau, R \cup \{\ulcorner i, j \urcorner\})$.

The shading lemma follows from this lemma.

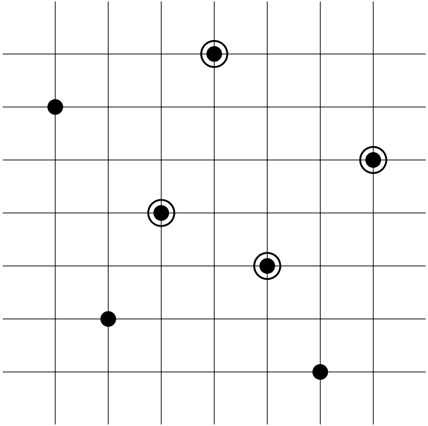
Comparing occurrences

Let $p = 2413$.



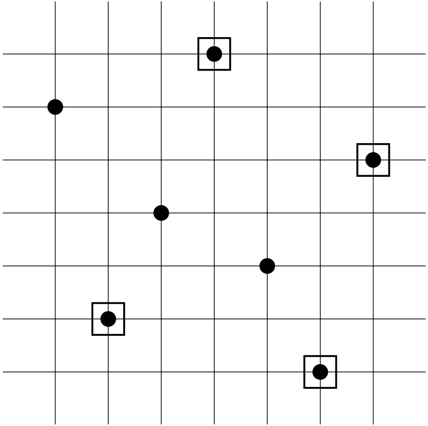
Comparing occurrences

Let $p = 2413$.



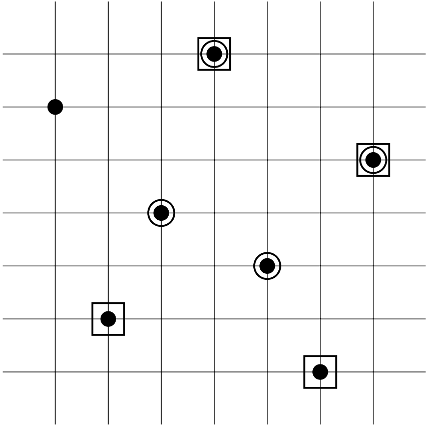
Comparing occurrences

Let $p = 2413$.



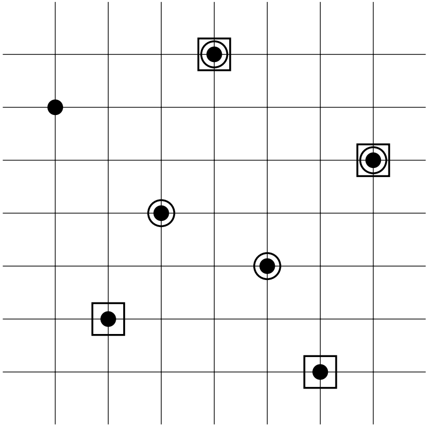
Comparing occurrences

Let $p = 2413$.



Comparing occurrences

Let $p = 2413$.



Force $F = ((4, N), (2, W), (1, W))$. The second (boxed) occurrence is stronger w.r.t. F.

Shading multiple boxes

Lemma 5

Let $p = (\tau, R)$ be a mesh pattern with force F, and $p' = (\tau, R')$ be another mesh pattern. Let $S = \{s_1, s_2, \dots, s_k\}$ where $S = R' \backslash R$. If all the sets

$$(\tau, R)^{s_1\star}, (\tau, R \cup \{s_1\})^{s_2\star}, \dots, (\tau, R \cup \{s_1, s_2, \dots, s_{k-1}\})^{s_k\star}$$

contain a non-trivial occurrence of p that is stronger than the trivial occurrence; or an occurrence of p', then containment of p implies containment of p'.

The simultaneous shading lemma follows from this lemma.

Results

Pattern	SL	Lemma 2	SSL	Lemma 5
12	237	237	229	221
123	34626	34618	34154	33634
132	34213	34213	33985	33621

Table: The number of coincidence classes found by different lemmas

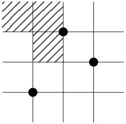
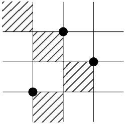
Generalized method

After fixing a force F, we can recursively apply a variant of the previous lemma to capture even more coincidences.

- Walk down a decision tree, branching on
  - box is empty,
  - box contains at least one point
- Leaves should give a contradiction, or an instance of the target pattern

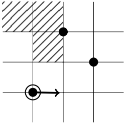
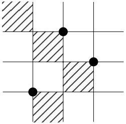
Example

- Prove that containment of p implies containment of p'

$p =$  $p' =$ 

Example

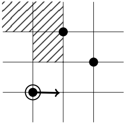
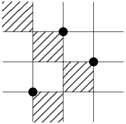
- Prove that containment of p implies containment of p'

$p =$  $p' =$ 

- Use force $F = ((1, E))$

Example

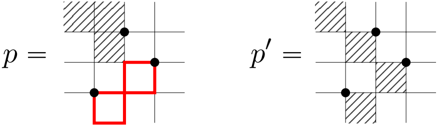
- Prove that containment of p implies containment of p'

$p =$  $p' =$ 

- Use force $F = ((1, E))$
- Consider a permutation π which contains p, and take the occurrence of p that has maximum strength w.r.t. F

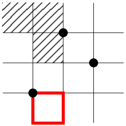
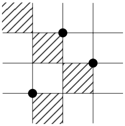
Example

- Want to add shading to $\ulcorner 1, 0 \urcorner$ and $\ulcorner 2, 1 \urcorner$



Example

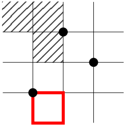
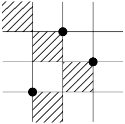
- Want to add shading to $\ulcorner\!\llcorner 1, 0 \lrcorner\!\urcorner$ and $\ulcorner\!\llcorner 2, 1 \lrcorner\!\urcorner$

$p =$  $p' =$ 

- Consider $\ulcorner\!\llcorner 1, 0 \lrcorner\!\urcorner$

Example

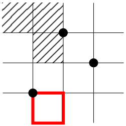
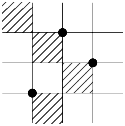
- Want to add shading to $\ulcorner 1, 0 \urcorner$ and $\ulcorner 2, 1 \urcorner$

$p =$  $p' =$ 

- Consider $\ulcorner 1, 0 \urcorner$
  - If empty, we're done

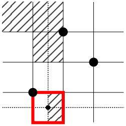
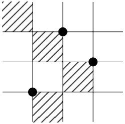
Example

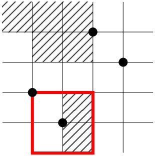
- Want to add shading to $\lceil\!\lfloor 1, 0 \rfloor\!\rceil$ and $\lceil\!\lfloor 2, 1 \rfloor\!\rceil$

$p =$  $p' =$ 

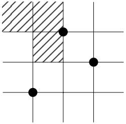
- Consider $\lceil\!\lfloor 1, 0 \rfloor\!\rceil$
  - If empty, we're done
  - Otherwise, consider the rightmost point

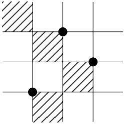
Example

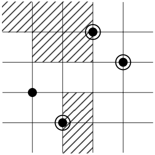
$p =$  $p' =$ 

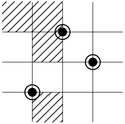
$w_1 = p^{\overrightarrow{\ulcorner 1, 0 \urcorner}} =$ 

Example

$p =$ 

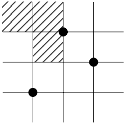
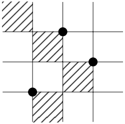
$p' =$ 

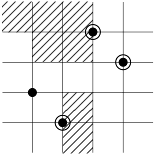
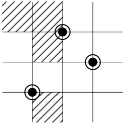
$w_1 = p^{\overrightarrow{\ulcorner 1, 0 \urcorner}} =$ 

$q_1 =$ 

- Consider subsequence at indices 234

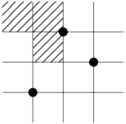
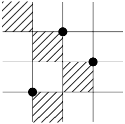
Example

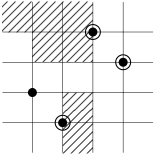
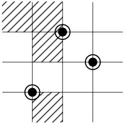
$p =$  $p' =$ 

$w_1 = p^{\overrightarrow{\ulcorner 1, 0 \urcorner}} =$  $q_1 =$ 

- Consider subsequence at indices 234
- q_1 is an occurrence of p that is stronger, a contradiction

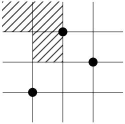
Example

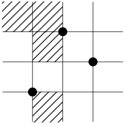
$p =$  $p' =$ 

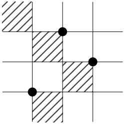
$w_1 = p^{\overrightarrow{\ulcorner 1, 0 \urcorner}} =$  $q_1 =$ 

- Consider subsequence at indices 234
- q_1 is an occurrence of p that is stronger, a contradiction
- Can assume that $\ulcorner 1, 0 \urcorner$ is empty

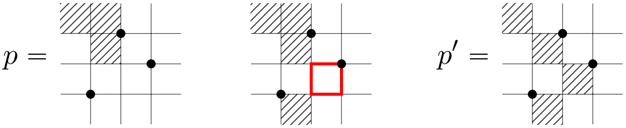
Example

$p =$ 



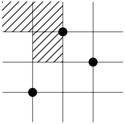
$p' =$ 

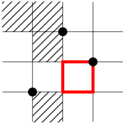
Example

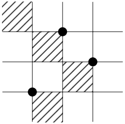


- Now consider $[2, 1]$

Example

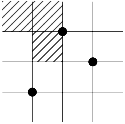
$p =$ 

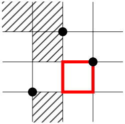


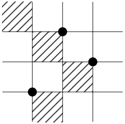
$p' =$ 

- Now consider $[2, 1]$
  - If empty, we're done

Example

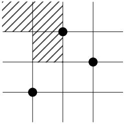
$p =$ 

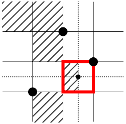


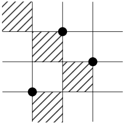
$p' =$ 

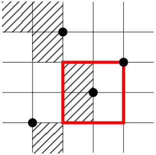
- Now consider $\ulcorner 2, 1 \urcorner$
  - If empty, we're done
  - Otherwise, consider the leftmost point

Example

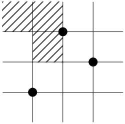
$p =$ 

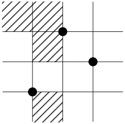


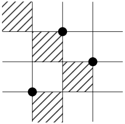
$p' =$ 

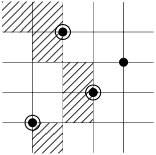
$w_2 =$ 

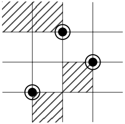
Example

$p =$ 



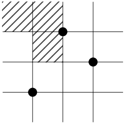
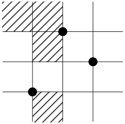
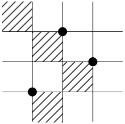
$p' =$ 

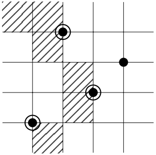
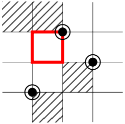
$w_2 =$ 

$q_2 =$ 

- Consider subsequence at indices 123

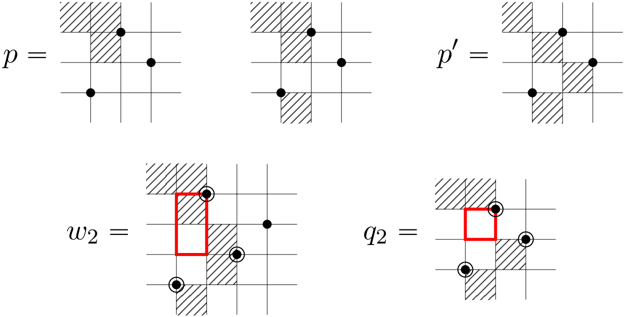
Example

$p =$   $p' =$ 

$w_2 =$  $q_2 =$ 

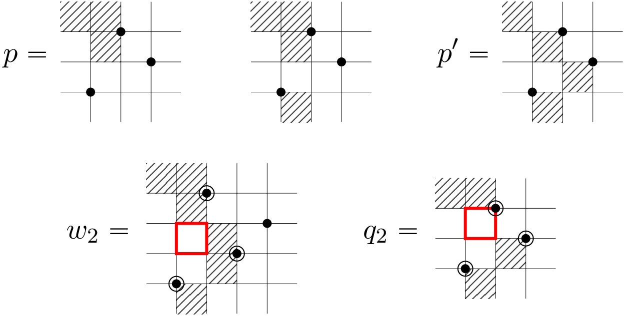
- Consider subsequence at indices 123
- Looking at q_2, we're missing $\ulcorner 1, 2 \urcorner$

Example



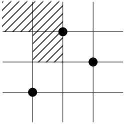
- Consider subsequence at indices 123
- Looking at q_2, we're missing $[1,2]$
- Corresponds to $[1,2]$ and $[1,3]$ in w_2

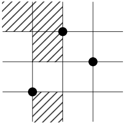
Example

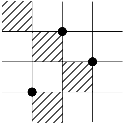


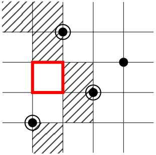
- Consider subsequence at indices 123
- Looking at q_2, we're missing $[1, 2]$
- Corresponds to $[1, 2]$ in w_2

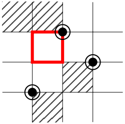
Example

$p =$ 



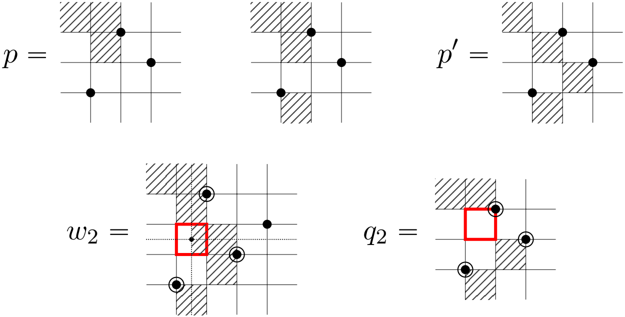
$p' =$ 

$w_2 =$ 

$q_2 =$ 

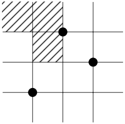
- Consider subsequence at indices 123
- Looking at q_2, we're missing $\llcorner\!\ulcorner 1, 2 \urcorner\!\lrcorner$
- Corresponds to $\llcorner\!\ulcorner 1, 2 \urcorner\!\lrcorner$ in w_2
    - If empty, we're done

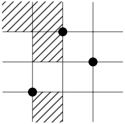
Example

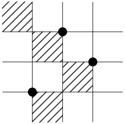


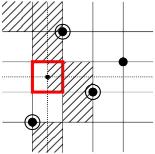
- Consider subsequence at indices 123
- Looking at q_2, we're missing $[1, 2]$
- Corresponds to $[1, 2]$ in w_2
  - If empty, we're done
  - Otherwise, consider the rightmost point

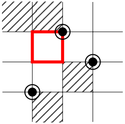
Example

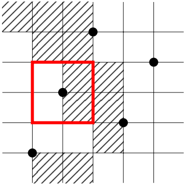
$p =$ 



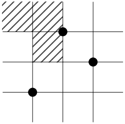
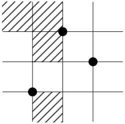
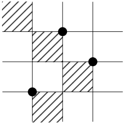
$p' =$ 

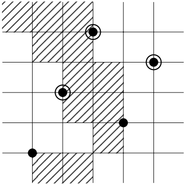
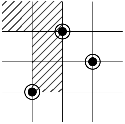
$w_2 =$ 

$q_2 =$ 

$w_3 =$ 

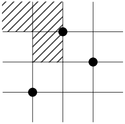
Example

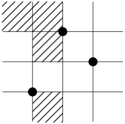
$p =$   $p' =$ 

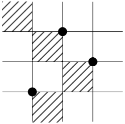
$w_3 =$  $q_3 =$ 

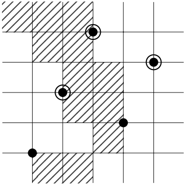
- Consider subsequence at indices 235

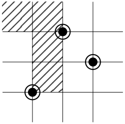
Example

$p =$ 



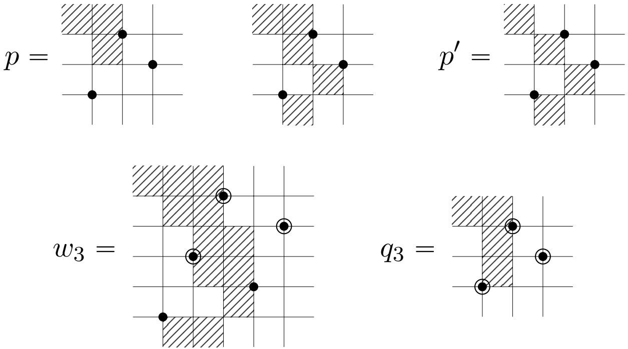
$p' =$ 

$w_3 =$ 

$q_3 =$ 

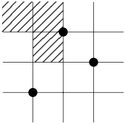
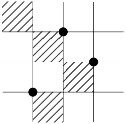
- Consider subsequence at indices 235
- q_3 is an occurrence of p that is stronger, a contradiction

Example



- Consider subsequence at indices 235
- q_3 is an occurrence of p that is stronger, a contradiction
- Can assume that $\ulcorner 2, 1 \lrcorner$ is empty as well

Example

$p =$  $p' =$ 

- We've proved that the occurrence with maximum strength w.r.t. F has no points in $\ulcorner 1, 0 \lrcorner$, $\ulcorner 2, 1 \lrcorner$
- Hence this is an occurrence of p' as well $\square$

Algorithm

We have formalized this into an algorithm (see abstract) and implemented it in Python. It is available on GitHub:

`http://tinyurl.com/shadingalgorithm`

- Currently not very user friendly
- Exhaustively search for proofs that recurse no more than d steps
- Has some optimizations, like deriving a force retroactively

Results

Let d denote the maximum recursion depth used in the proof.

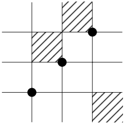
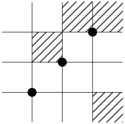
Pattern	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$
123	33634	33602	33548	33540	33538	33536	?
132	33621	33459	33412	33395	33394	33394	33390

Table: The number of coincidence classes found by different depths

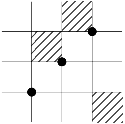
Full coincidence classification of length 3 mesh patterns

After dealing with a handful of special cases we show that the number of coincidence classes for the patterns in Table 2 are 33516 and 33350, respectively.

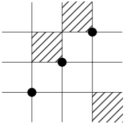
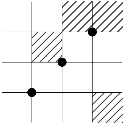
One of those special cases

$p =$  $p' =$ 

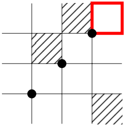
- Let force $F = ((1,S),(2,W),(3,W))$



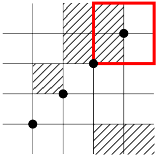
One of those special cases

$p =$  $p' =$ 

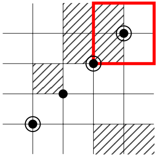
- Let force $F = ((1, S), (2, W), (3, W))$



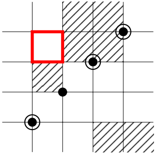
One of those special cases



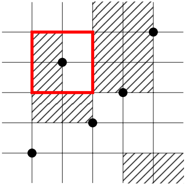
One of those special cases



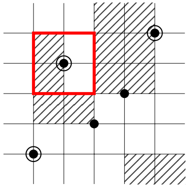
One of those special cases



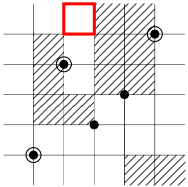
One of those special cases



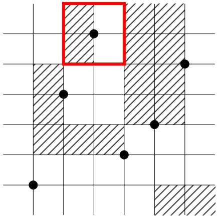
One of those special cases



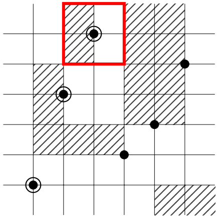
One of those special cases



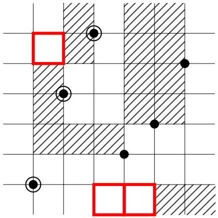
One of those special cases



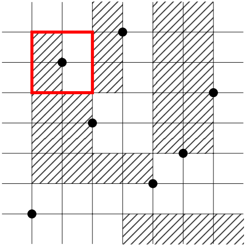
One of those special cases



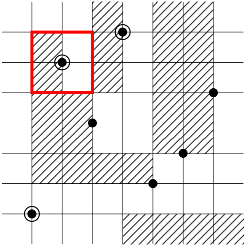
One of those special cases



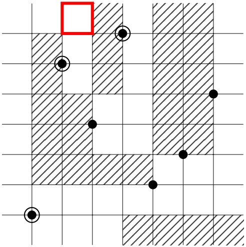
One of those special cases



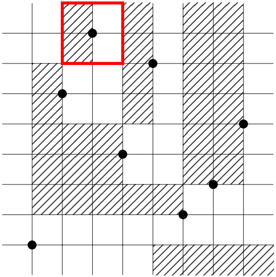
One of those special cases



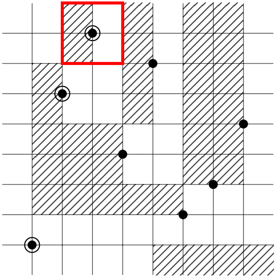
One of those special cases



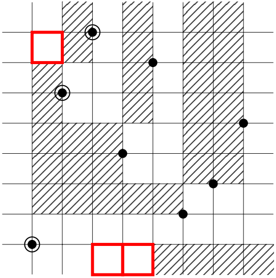
One of those special cases



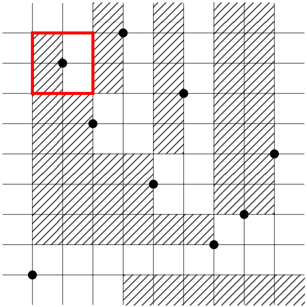
One of those special cases



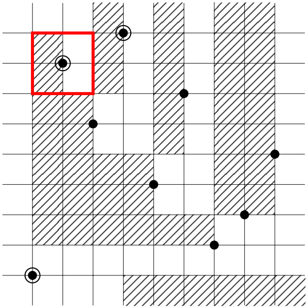
One of those special cases



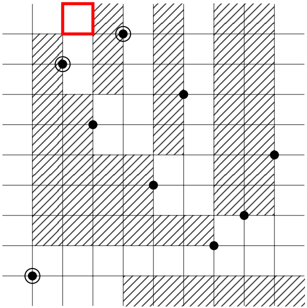
One of those special cases



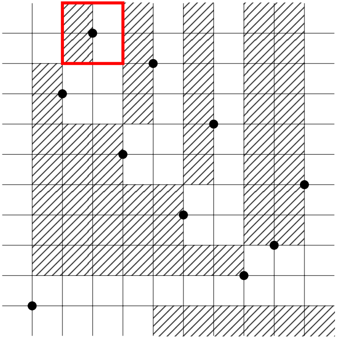
One of those special cases



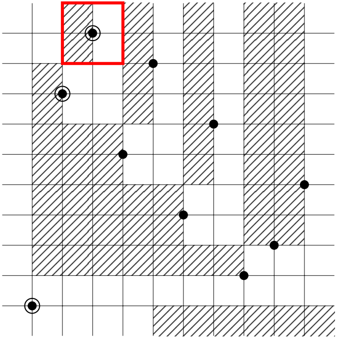
One of those special cases



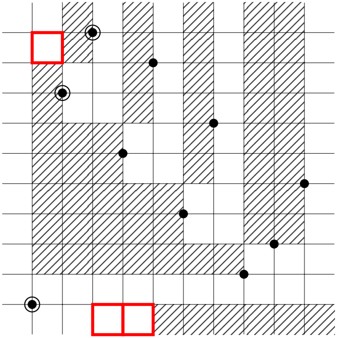
One of those special cases



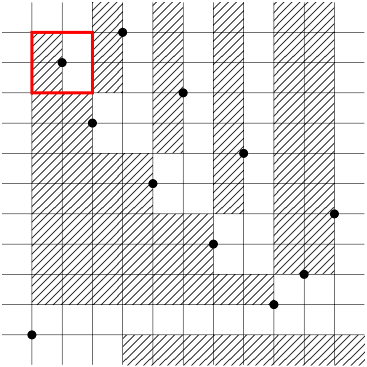
One of those special cases



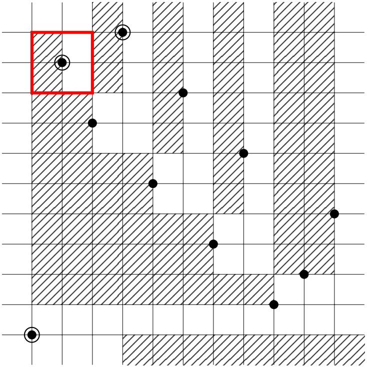
One of those special cases



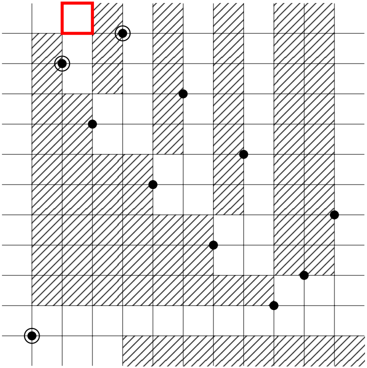
One of those special cases



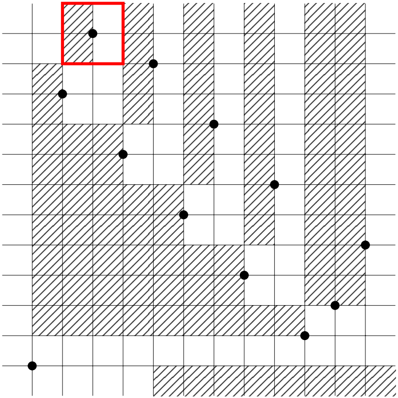
One of those special cases



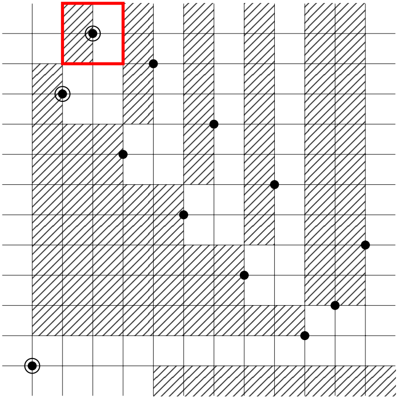
One of those special cases



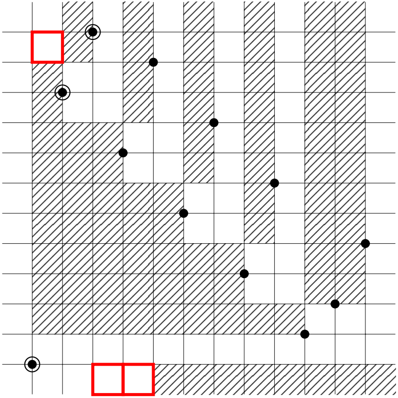
One of those special cases



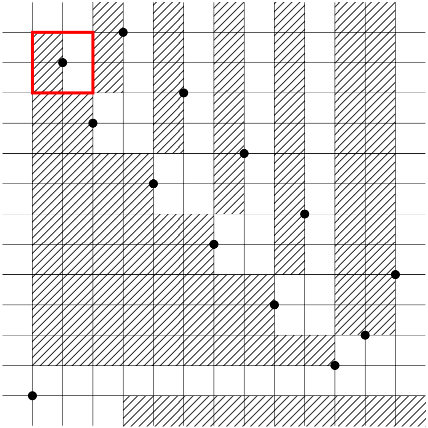
One of those special cases



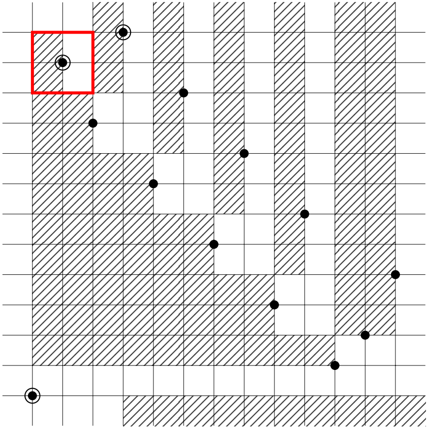
One of those special cases



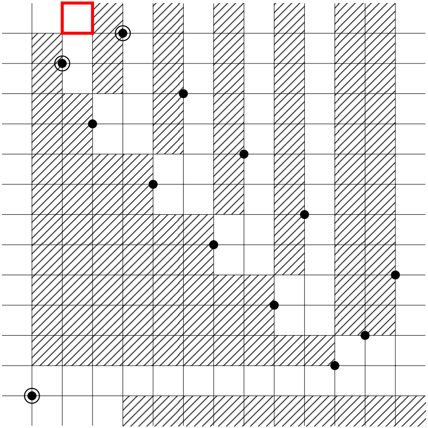
One of those special cases



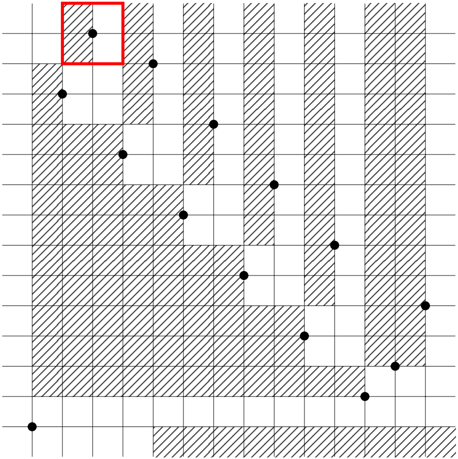
One of those special cases



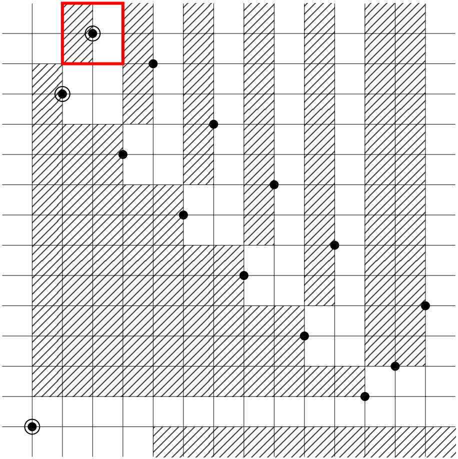
One of those special cases



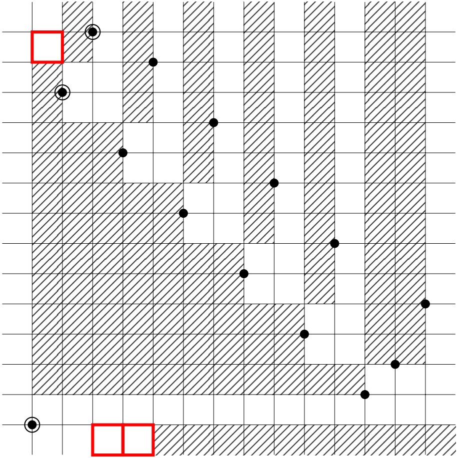
One of those special cases



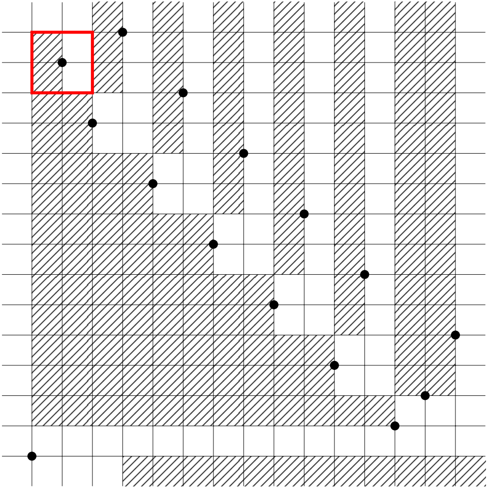
One of those special cases



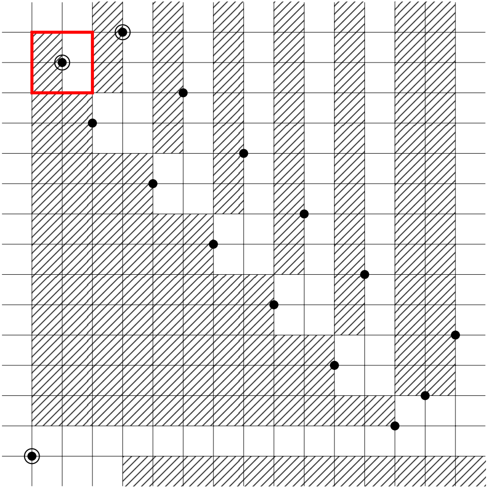
One of those special cases



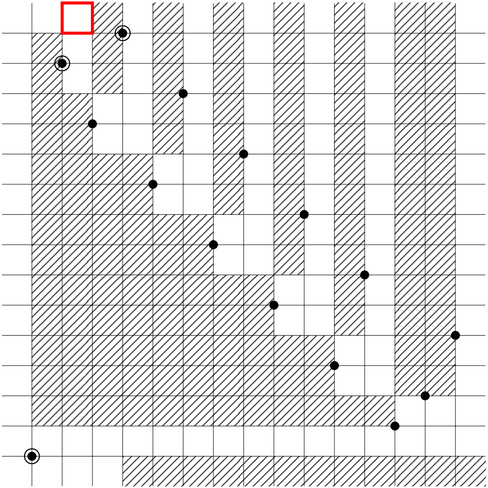
One of those special cases



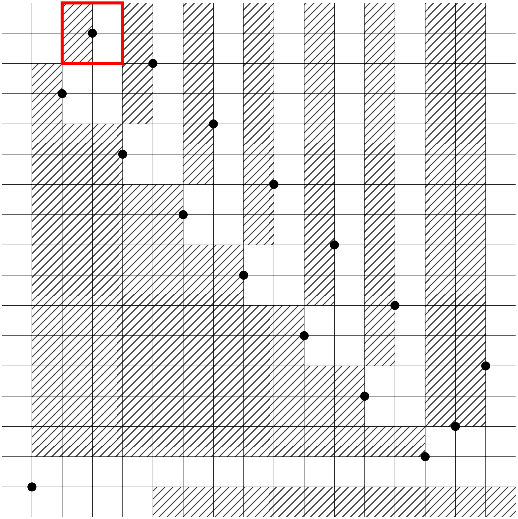
One of those special cases



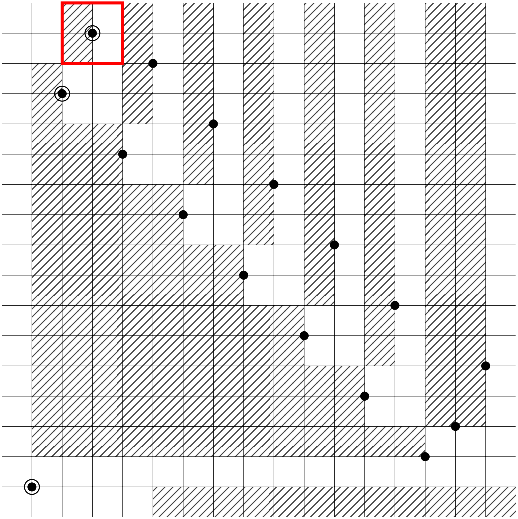
One of those special cases



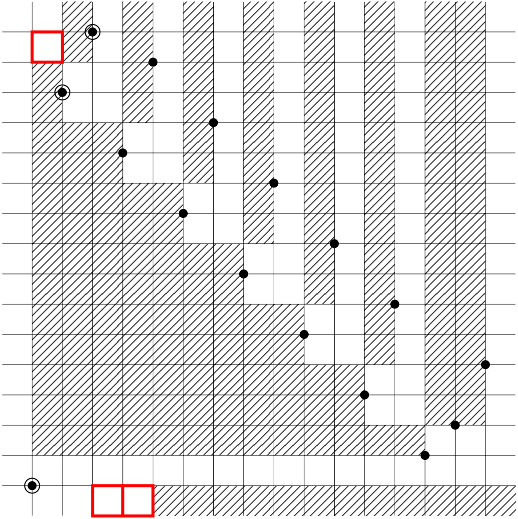
One of those special cases



One of those special cases



One of those special cases



Example

For a finite permutation, this process will eventually exhaust all points. When that happens, we must have found a new occurrence that is stronger w.r.t. F, a contradiction.

Future work

- Find a way to handle “infinite” cases neatly
- Support more complex conditions than just a point in a box, e.g.
    - point in either of two boxes,
    - inversion across two consecutive boxes
- Use unidirectional containment to prove coincidences
- Classify length 4 patterns

Thanks for listening!

Claesson, Tenner, and Ulfarsson.
Coincidence among families of mesh patterns.
Australas. J. Combin., 2015.

Hilmarsson, Jónsdóttir, Sigurðardóttir, Viðarsdóttir, and Ulfarsson.
Wilf-classification of mesh patterns of short length.
Electron. J. Combin, 2015.