

Automatic discovery of polynomial time enumerations

Unnar Freyr Erlendsson

Joint work with Christian Bean, Jay Pantone and Henning Ulfarsson

Permutation Patterns 2019



Gridded permutations

A *gridded permutation* is a permutation along with a list of cells we refer to as the *gridding* of the permutation.

Gridded permutations

A *gridded permutation* is a permutation along with a list of cells we refer to as the *gridding* of the permutation.

Containment and avoidance is defined as with regular permutations except it also requires the griddings to be equal.

TileScope algorithm

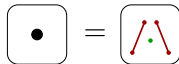
The TileScope algorithm developed by Bean [1] built ontop of CombSpecSearcher works with *Tilings* to find structure in permutation classes.

TileScope algorithm

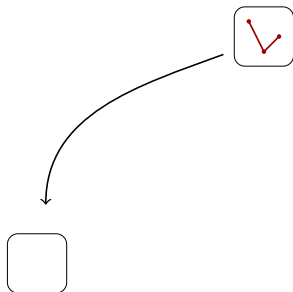
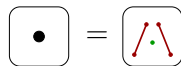
The TileScope algorithm developed by Bean [1] built on top of CombSpecSearcher works with *Tilings* to find structure in permutation classes.

If it succeeds in finding a structural description of the permutation class it returns something we call a *proof tree*.

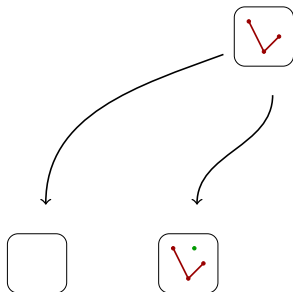
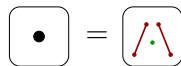
Proof tree $Av(312)$



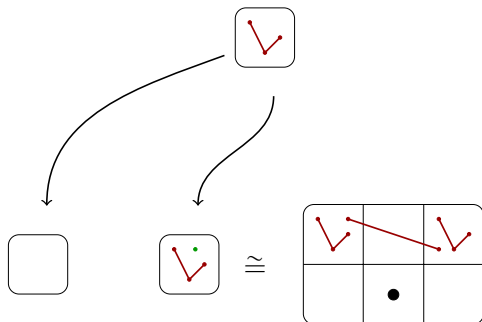
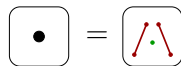
Proof tree Av(312)



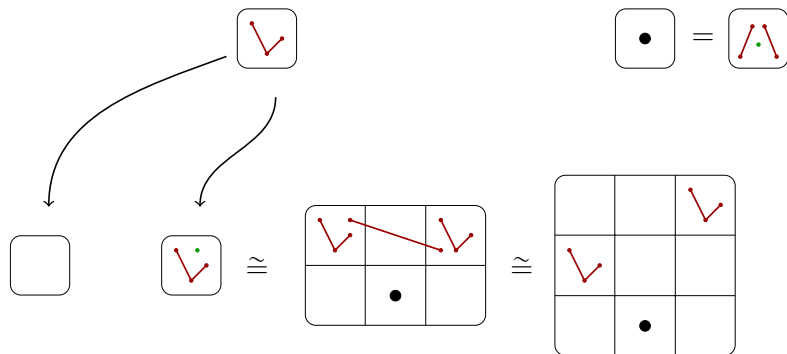
Proof tree $\text{Av}(312)$



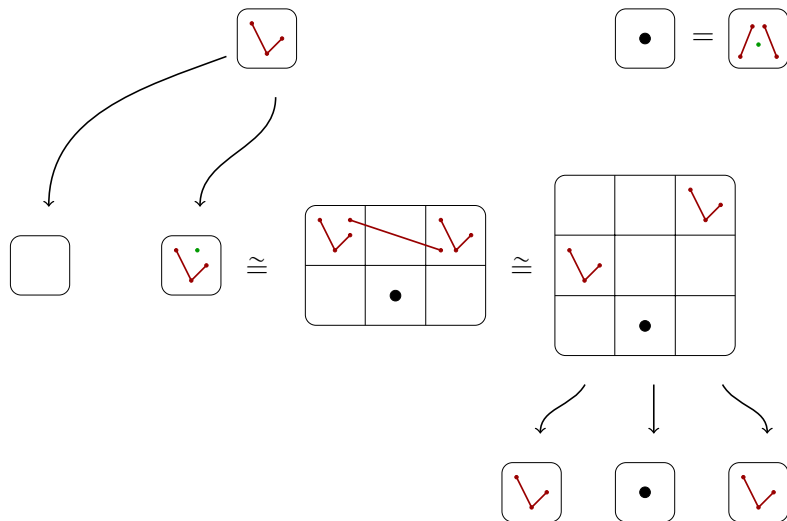
Proof tree Av(312)



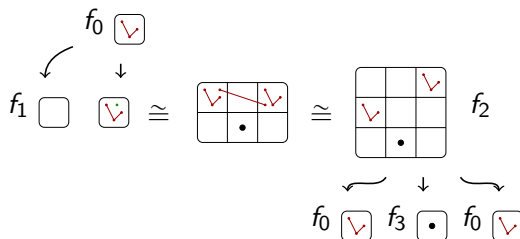
Proof tree Av(312)



Proof tree Av(312)



Enumerating proof trees



We can write down a recurrence relation from the structural description

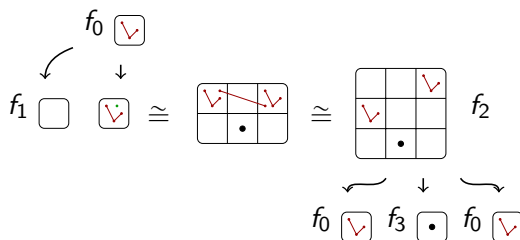
$$f_0(n) = f_1(n) + f_2(n)$$

$$f_1(n) = 1 \text{ if } n = 0 \text{ else } 0$$

$$f_2(n) = \sum_{i=0}^n \sum_{j=0}^{n-i} f_0(i) \cdot f_3(j) \cdot f_0(n-i-j)$$

$$f_3(n) = 1 \text{ if } n = 1 \text{ else } 0$$

Enumerating proof trees



We can write down a recurrence relation from the structural description

$$f_0(n) = f_1(n) + f_2(n)$$

$$f_1(n) = 1 \text{ if } n = 0 \text{ else } 0$$

$$f_2(n) = \sum_{i=0}^{n-1} f_0(i) \cdot f_0(n-i-1)$$

$$f_3(n) = 1 \text{ if } n = 1 \text{ else } 0$$

The core idea of CombSpecSearcher is to repeatedly apply strategies to combinatorial classes to obtain structural descriptions.

The core idea of CombSpecSearcher is to repeatedly apply strategies to combinatorial classes to obtain structural descriptions. The most common strategies fall under these categories

The core idea of CombSpecSearcher is to repeatedly apply strategies to combinatorial classes to obtain structural descriptions. The most common strategies fall under these categories

- Equivalence: $A = B$

The core idea of CombSpecSearcher is to repeatedly apply strategies to combinatorial classes to obtain structural descriptions. The most common strategies fall under these categories

- Equivalence: $A = B$
- Batch: $A = B_0 \sqcup B_1 \sqcup \cdots \sqcup B_k$

The core idea of CombSpecSearcher is to repeatedly apply strategies to combinatorial classes to obtain structural descriptions. The most common strategies fall under these categories

- Equivalence: $A = B$
- Batch: $A = B_0 \sqcup B_1 \sqcup \cdots \sqcup B_k$
- Decomposition: $A = B_0 \times B_1 \times \cdots \times B_k$

The core idea of CombSpecSearcher is to repeatedly apply strategies to combinatorial classes to obtain structural descriptions. The most common strategies fall under these categories

- Equivalence: $A = B$
- Batch: $A = B_0 \sqcup B_1 \sqcup \cdots \sqcup B_k$
- Decomposition: $A = B_0 \times B_1 \times \cdots \times B_k$
- Verification: we can directly count A

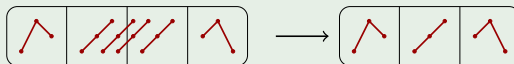
Fusion strategy

Two rows (or columns) are fusable if and only if it does not change the set of griddable permutations on the tiling.

Fusion strategy

Two rows (or columns) are fusable if and only if it does not change the set of griddable permutations on the tiling.

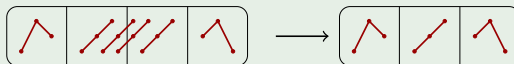
Example (Fusable tiling)



Fusion strategy

Two rows (or columns) are fusable if and only if it does not change the set of griddable permutations on the tiling.

Example (Fusable tiling)



Example (Not a fusable tiling)



Enumerating the fusion operator

To enumerate the fusion strategy we need to be able to track the number of points in the row (or column) that was fused.

Enumerating the fusion operator

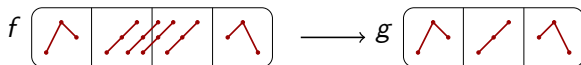
To enumerate the fusion strategy we need to be able to track the number of points in the row (or column) that was fused.

This can be done with a multivariate generating function or recurrence relation.

Enumerating the fusion operator

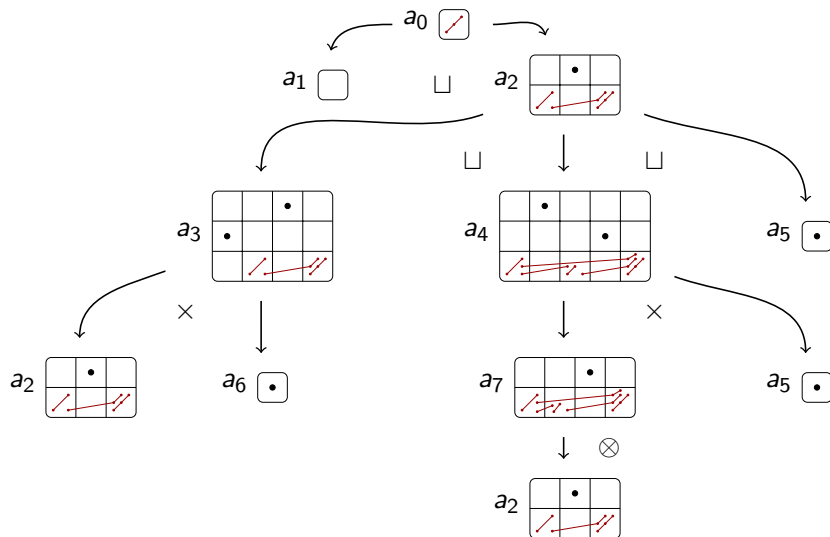
To enumerate the fusion strategy we need to be able to track the number of points in the row (or column) that was fused.

This can be done with a multivariate generating function or recurrence relation.

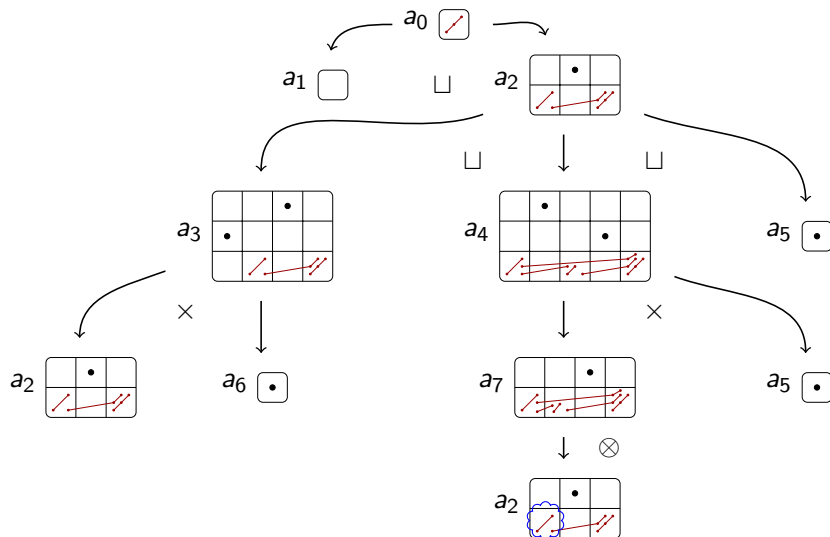


$$f(n) = \sum_{i=0} (i+1) \cdot g(n, i)$$

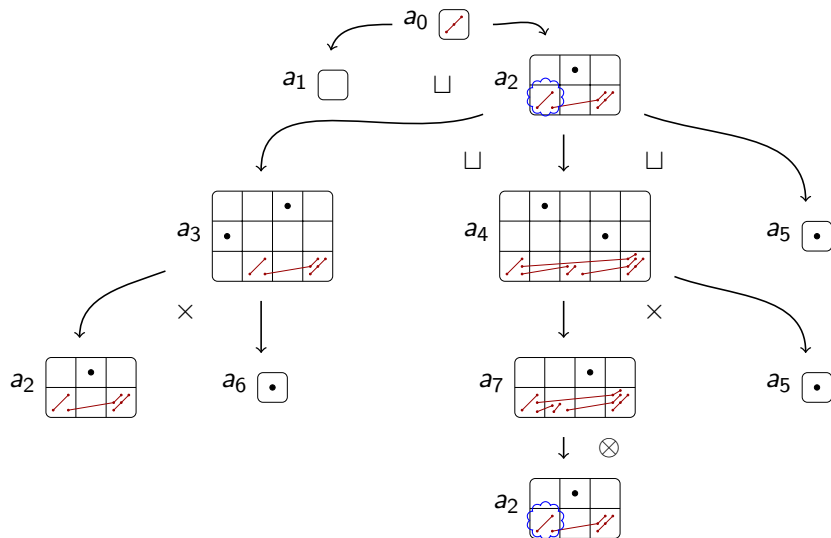
Enumerating $Av(123)$ using fusion



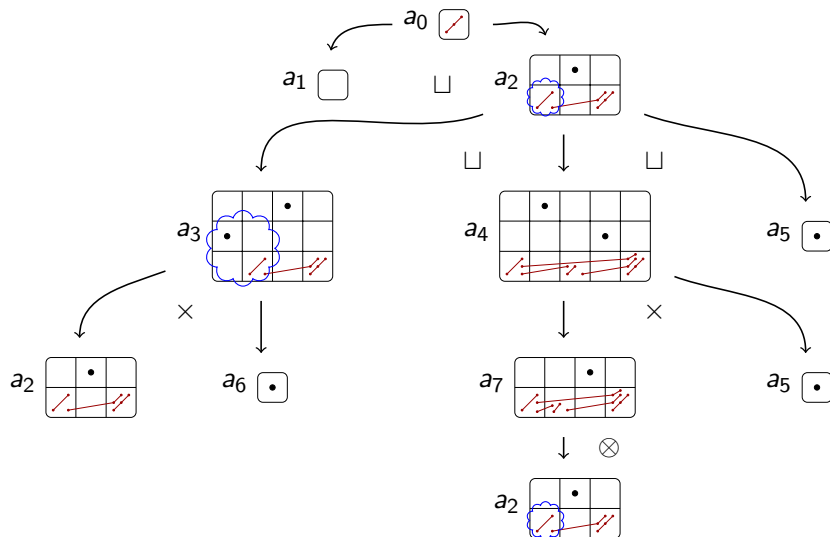
Enumerating $Av(123)$ using fusion



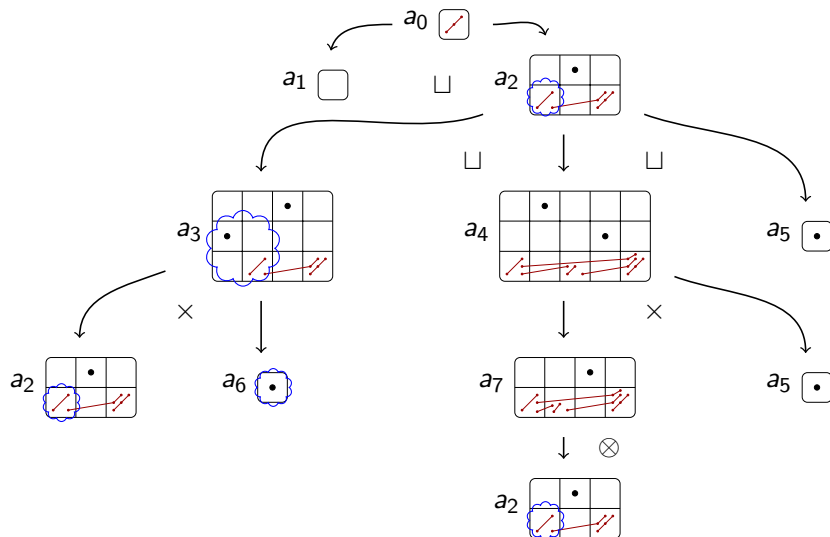
Enumerating $\text{Av}(123)$ using fusion



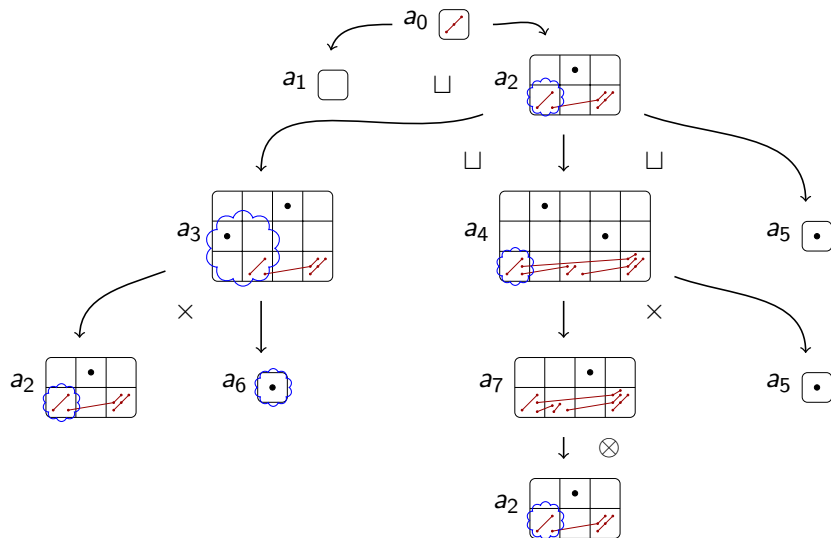
Enumerating $A_v(123)$ using fusion



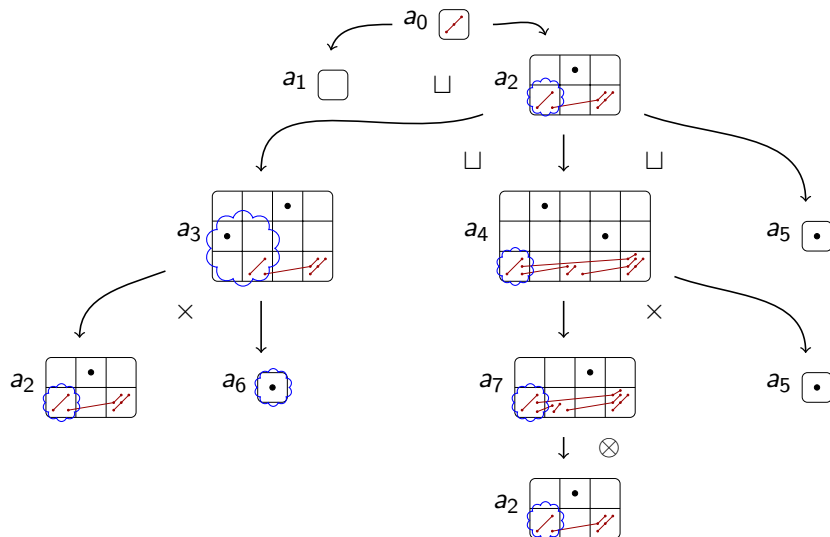
Enumerating $Av(123)$ using fusion



Enumerating $A_v(123)$ using fusion



Enumerating $A_v(123)$ using fusion



Fusion tree recurrence relation

$$a_0(n) = a_1(n) + \sum_{i=0}^n a_2(n, i)$$

$$a_1(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$a_2(n, i) = a_3(n, i) + a_4(n, i) + a_5(n, i)$$

$$a_3(n, i) = \sum_{m=0}^n \sum_{j=0}^i a_2(m, j) \cdot a_6(n - m, i - j)$$

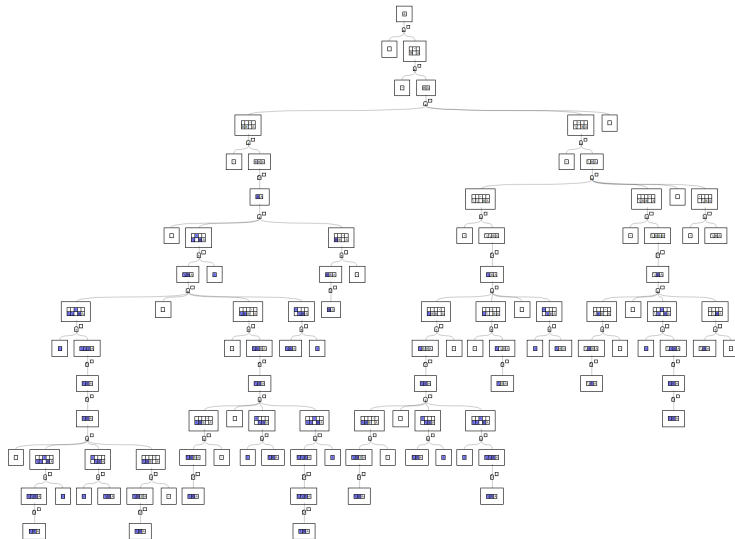
$$a_4(n, i) = \sum_{m=0}^n \sum_{j=0}^i a_7(m, j) \cdot a_5(n - m, i - j)$$

$$a_5(n, i) = \begin{cases} 1 & \text{if } n = 1 \text{ and } i = 0 \\ 0 & \text{otherwise} \end{cases}$$

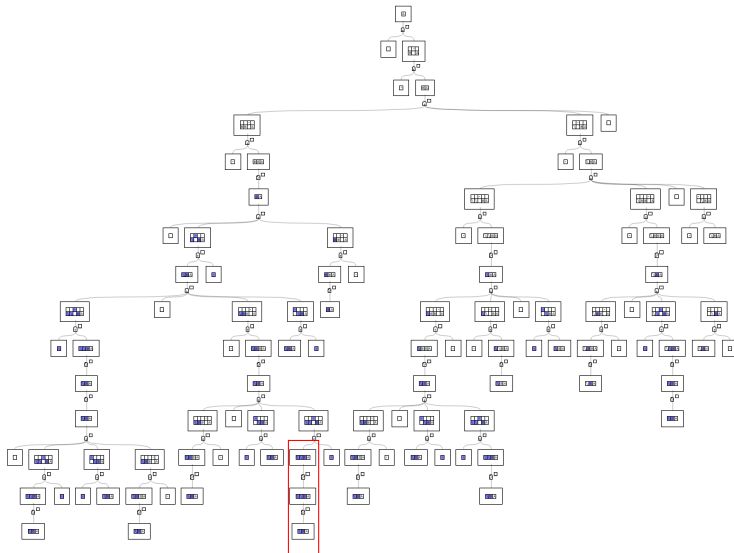
$$a_6(n, i) = \begin{cases} 1 & \text{if } n = 1 \text{ and } i = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$a_7(n, i) = \sum_{j=i}^n a_2(n, j)$$

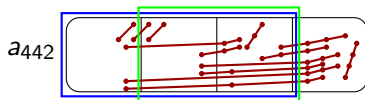
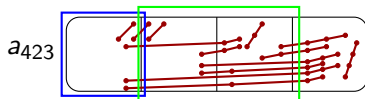
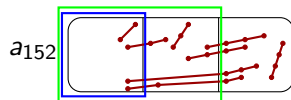
Enumerating $\text{Av}(1234)$



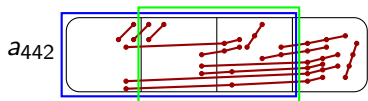
Enumerating $Av(1234)$



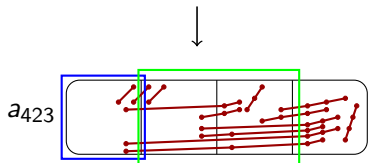
$Av(1234)$ example fusion



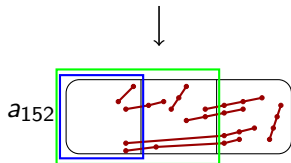
Av(1234) example fusion



$$a_{442}(n, b, g) = a_{423}(n, b - g, g)$$



$$a_{423}(n, b, g) = \sum_{l=b}^b \sum_{r=0}^{n-l} a_{152}(n, l + r, g + l)$$



Theorem (E.)

Guided search will terminate for any TileScope fusion tree.

TileScope fusion trees

Theorem (E.)

Guided search will terminate for any TileScope fusion tree.

Conjecture

Guided search will terminate successfully for any TileScope fusion tree.

Our success allowing the fusion strategy

$ B $	number of bases remaining	success with fusion
6	1	1
5	2	2
4	7	7
3	20	20
2	19	18
1	7	2
total	56	50

Table: The successes for bases consisting of length 4 patterns with fusion.

Overall we can get polynomial time enumeration algorithms for all bases consisting of length 4 patterns except 5 principal classes and $A_V(1432, 2143)$.

Conclusion

For more complicated strategies it can be easier to get a polynomial recurrence relation rather than a generating function which can be used to compute the terms of the sequence.

For more complicated strategies it can be easier to get a polynomial recurrence relation rather than a generating function which can be used to compute the terms of the sequence.

One can then try to guess generating functions that match the initial terms in order to do more analysis on the enumeration such as finding the growth rate or doing Wilf-classification.