

Mapped Tilings

A New Approach for Automatic Enumeration

Joint With

Christian Bean, Keele University

Abigail Olson, Keele University

Henning Ulfarsson, Reykjavik University





Outline

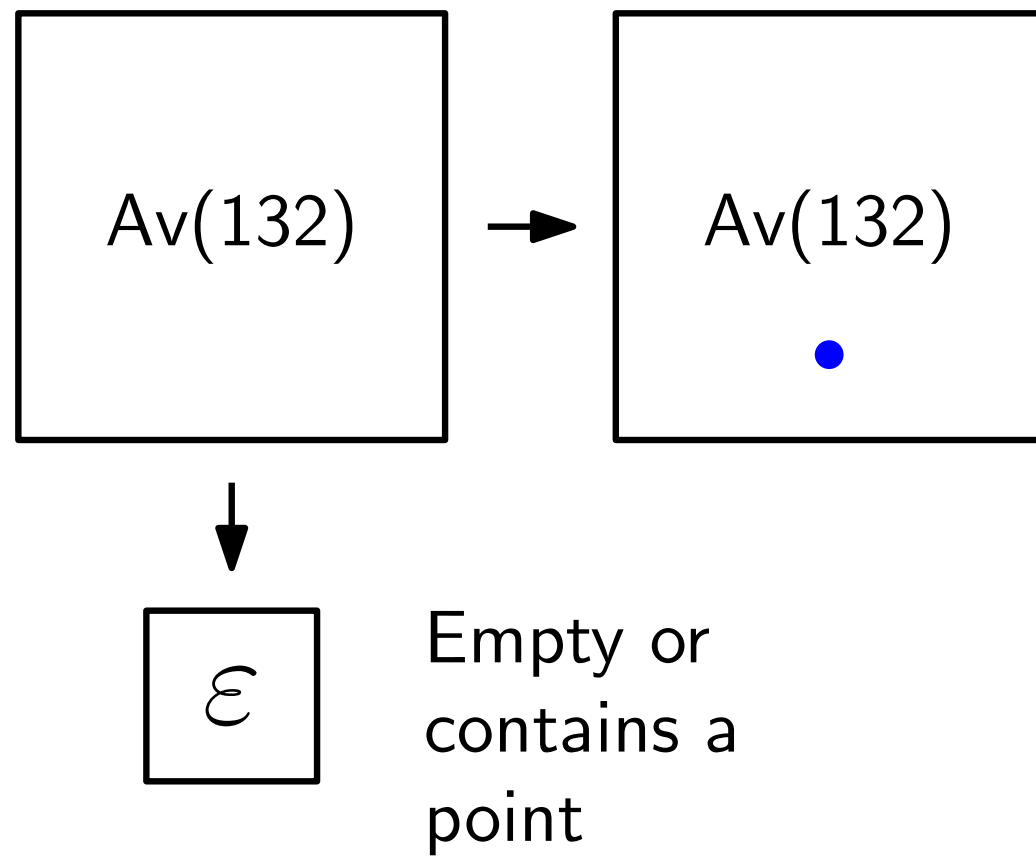
- Introduction to Tilings
- Telescope
- Parameters
- Mapped Tilings



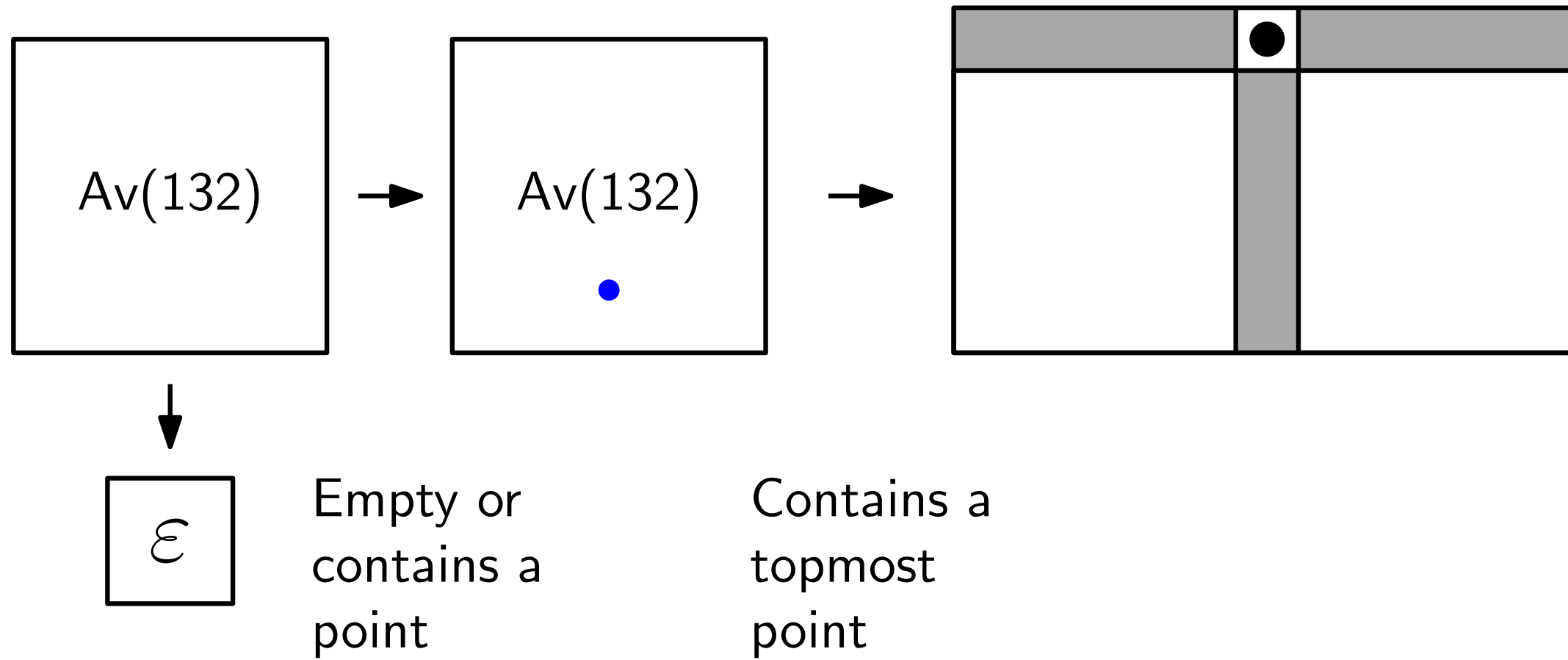
$Av(132)$

A Familiar Proof

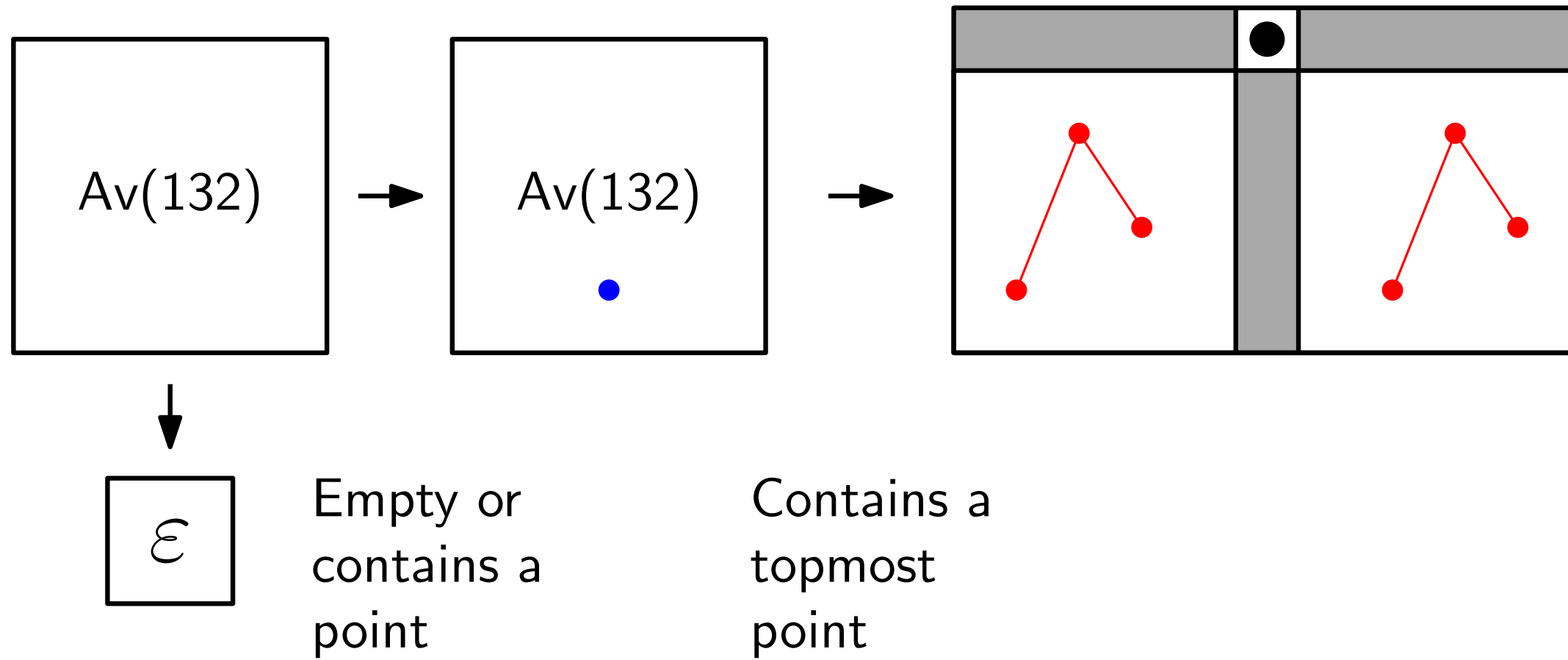
A Familiar Proof



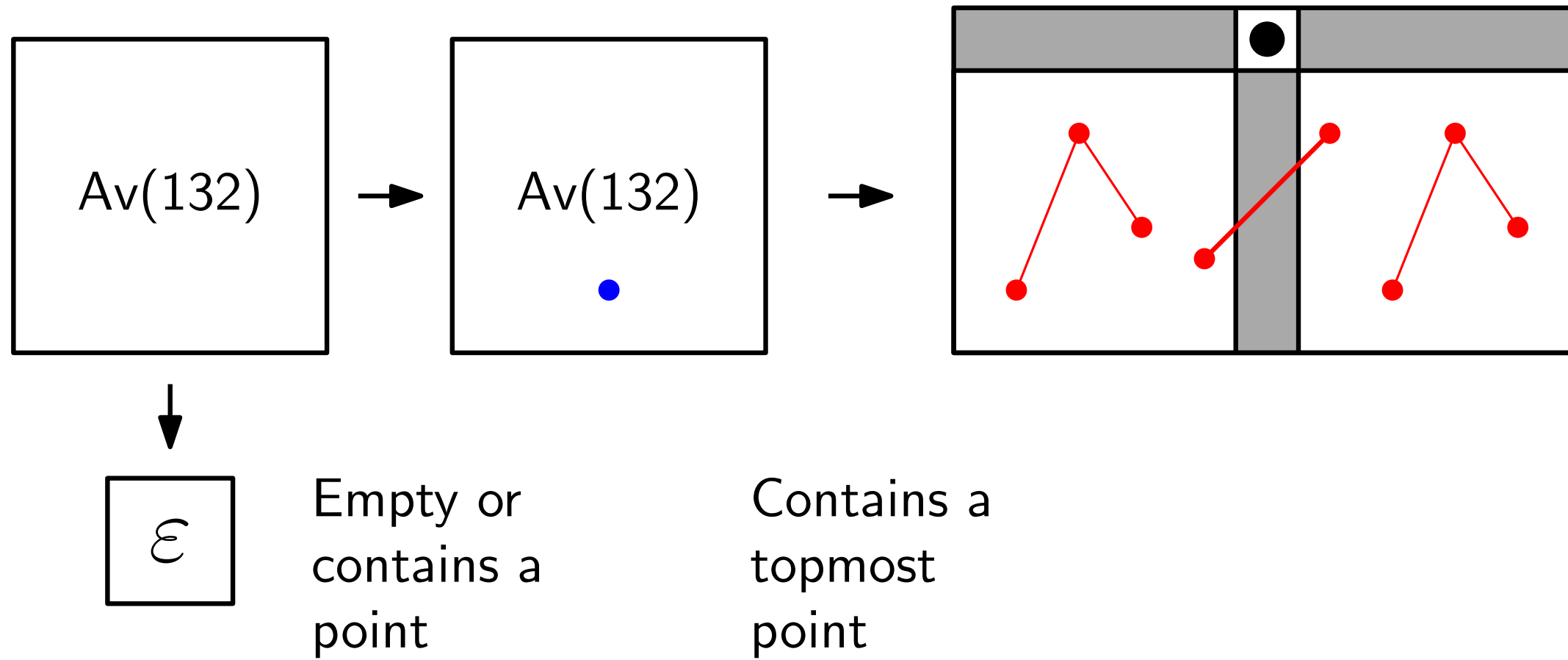
A Familiar Proof



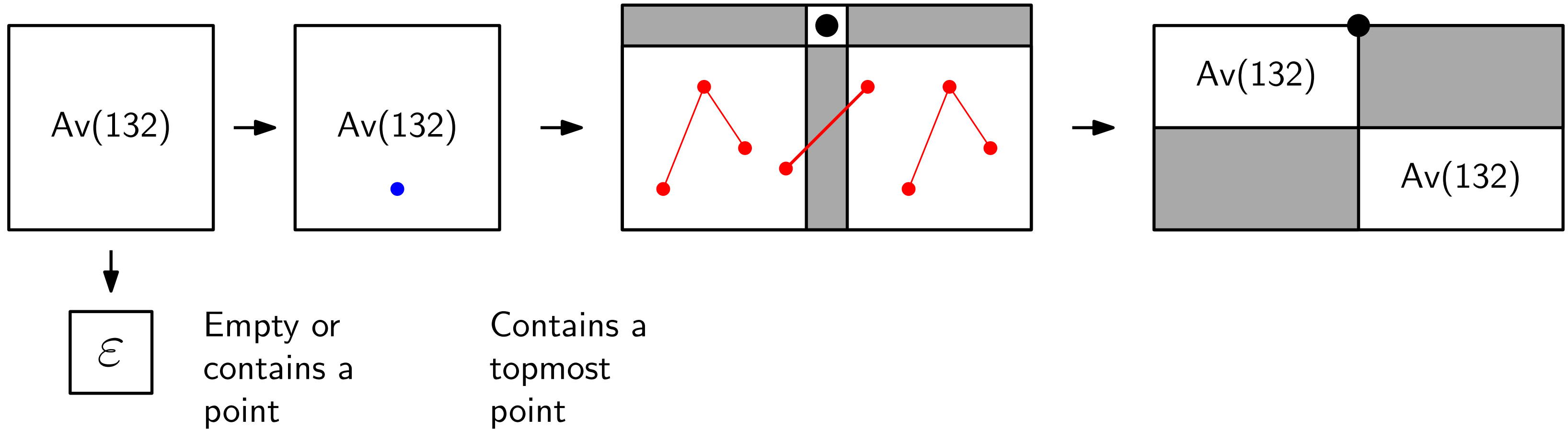
A Familiar Proof

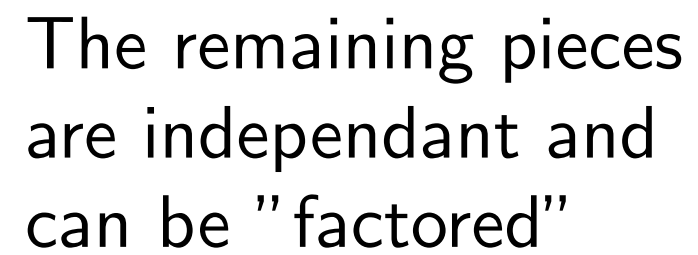


A Familiar Proof



A Familiar Proof



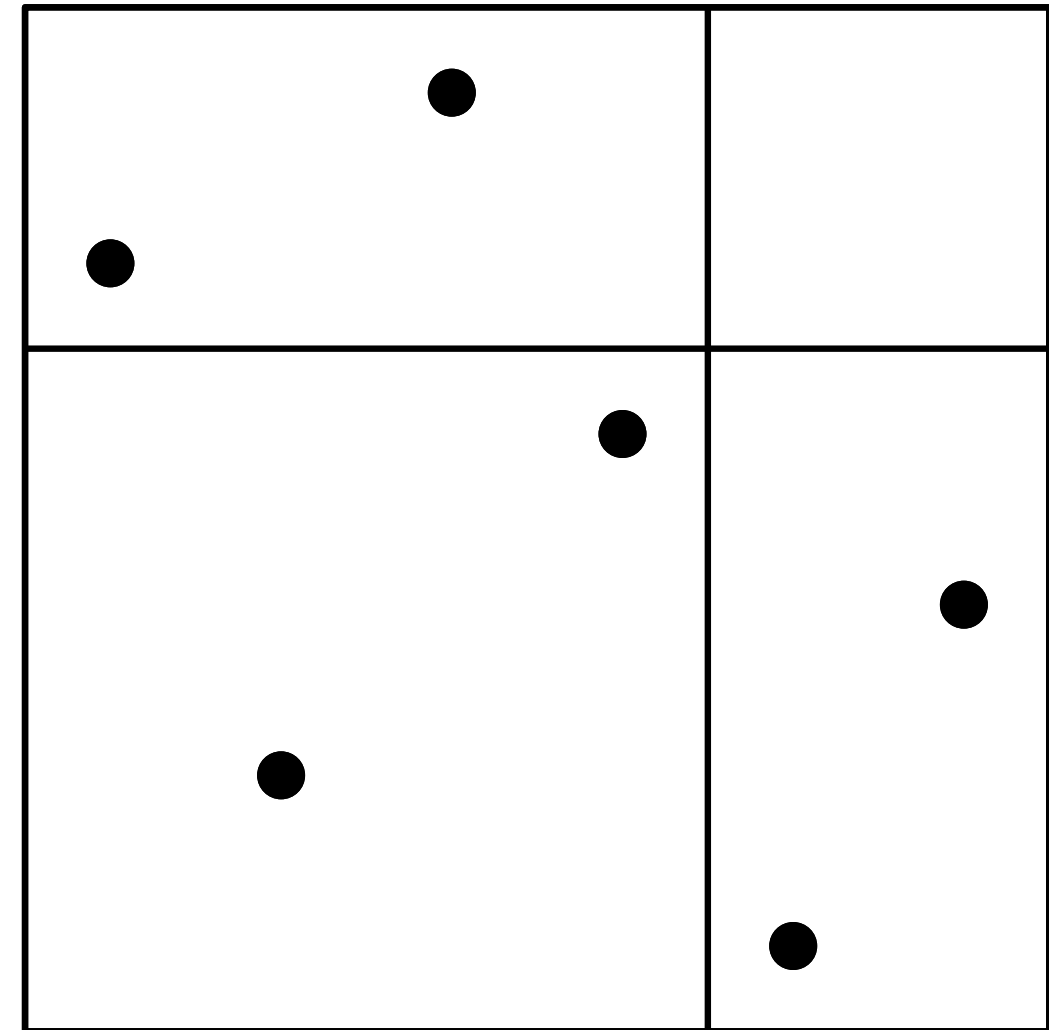




Introduction to Tilings

Definitions:

A **gridded permutation** is a pattern $\pi = \pi_1\pi_2 \dots \pi_n$ with a *valid* gridding $g = (x_1, y_1)(x_2, y_2) \dots (x_n, y_n)$.

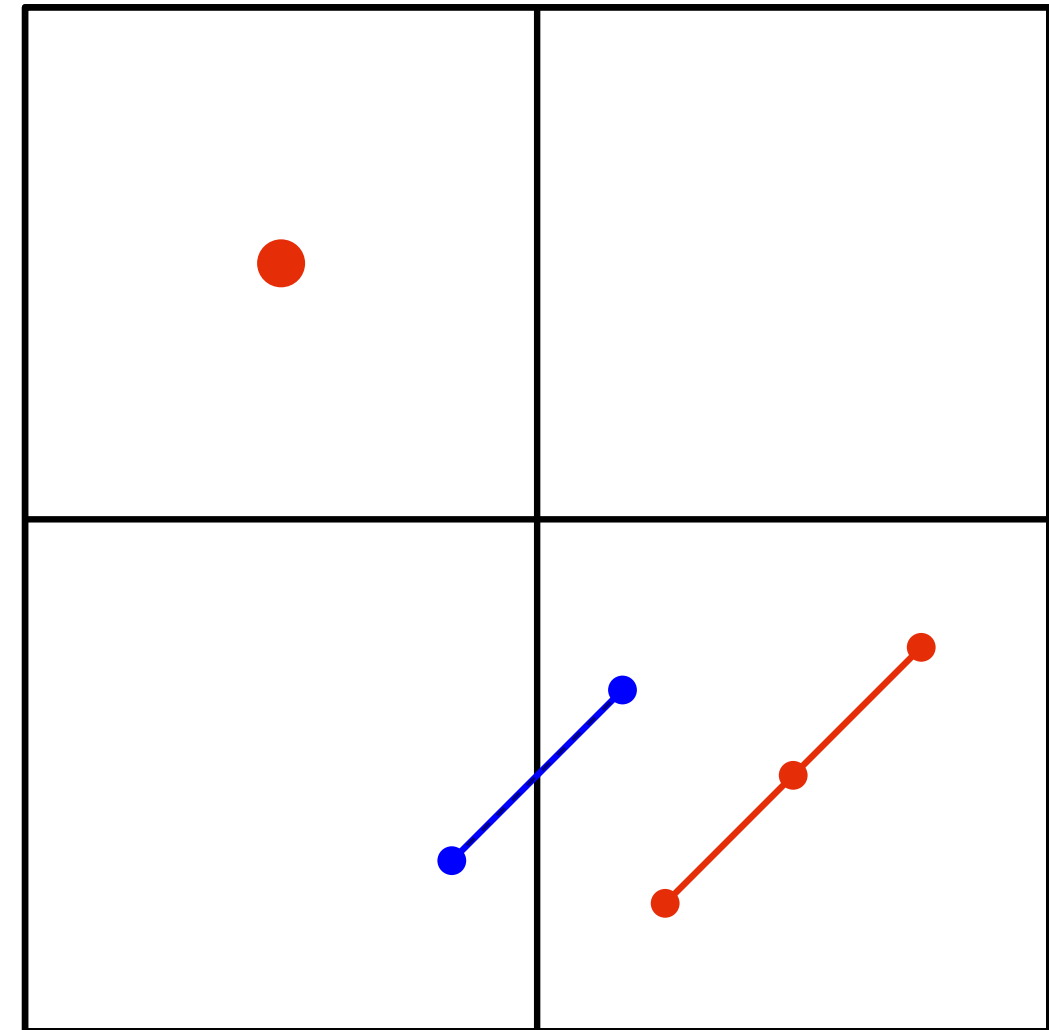


A gridded permutation with underlying pattern $\pi = 526413$ and gridding $g = (0, 1)(0, 0)(0, 1)(0, 0)(1, 0)(1, 0)$

Introduction to Tilings

Definitions:

A **tiling** defines a class of gridded permutations on a fixed grid using *obstructions* and *requirements*:



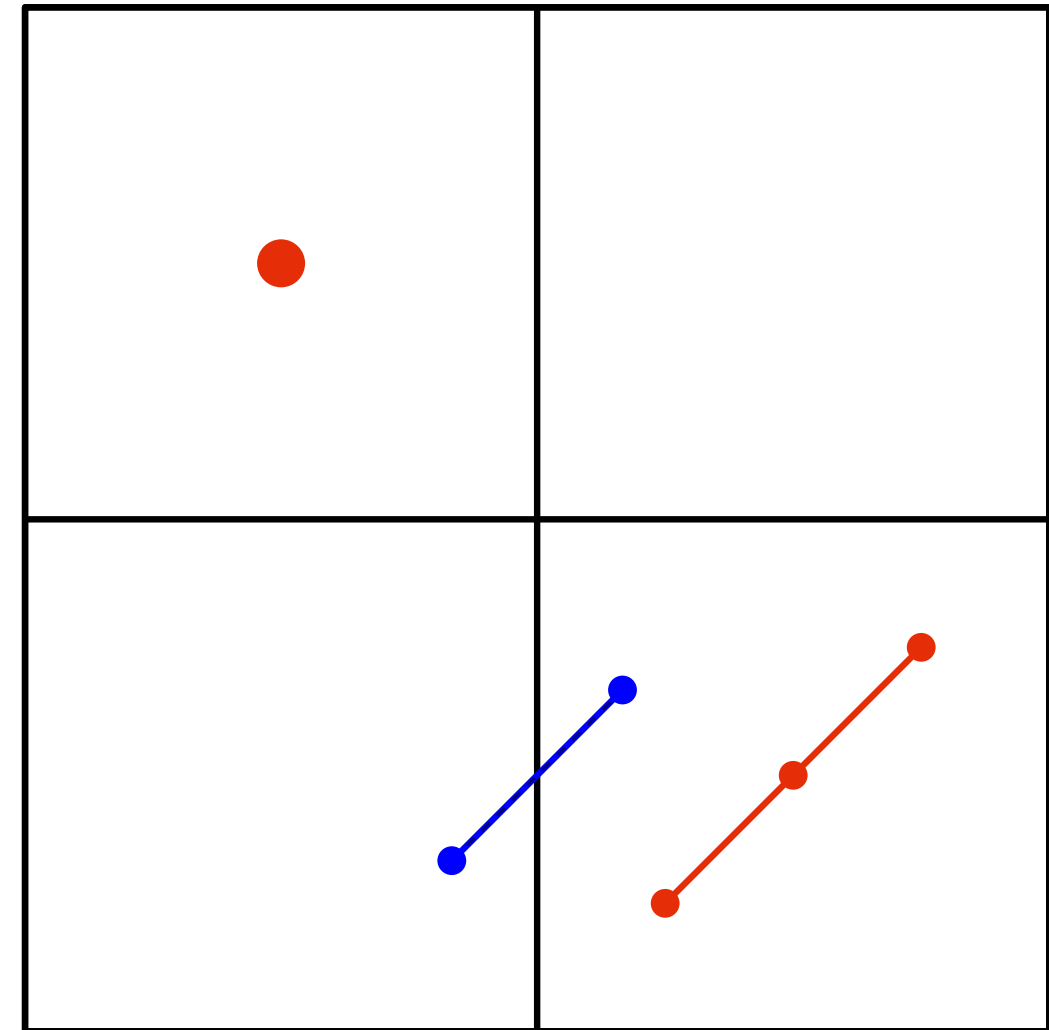
A tiling with obstructions and requirements.

Introduction to Tilings

Definitions:

A **tiling** defines a class of gridded permutations on a fixed grid using *obstructions* and *requirements*:

obstructions are gridded permutations that must be avoided by every member of the class



A tiling with obstructions and requirements.

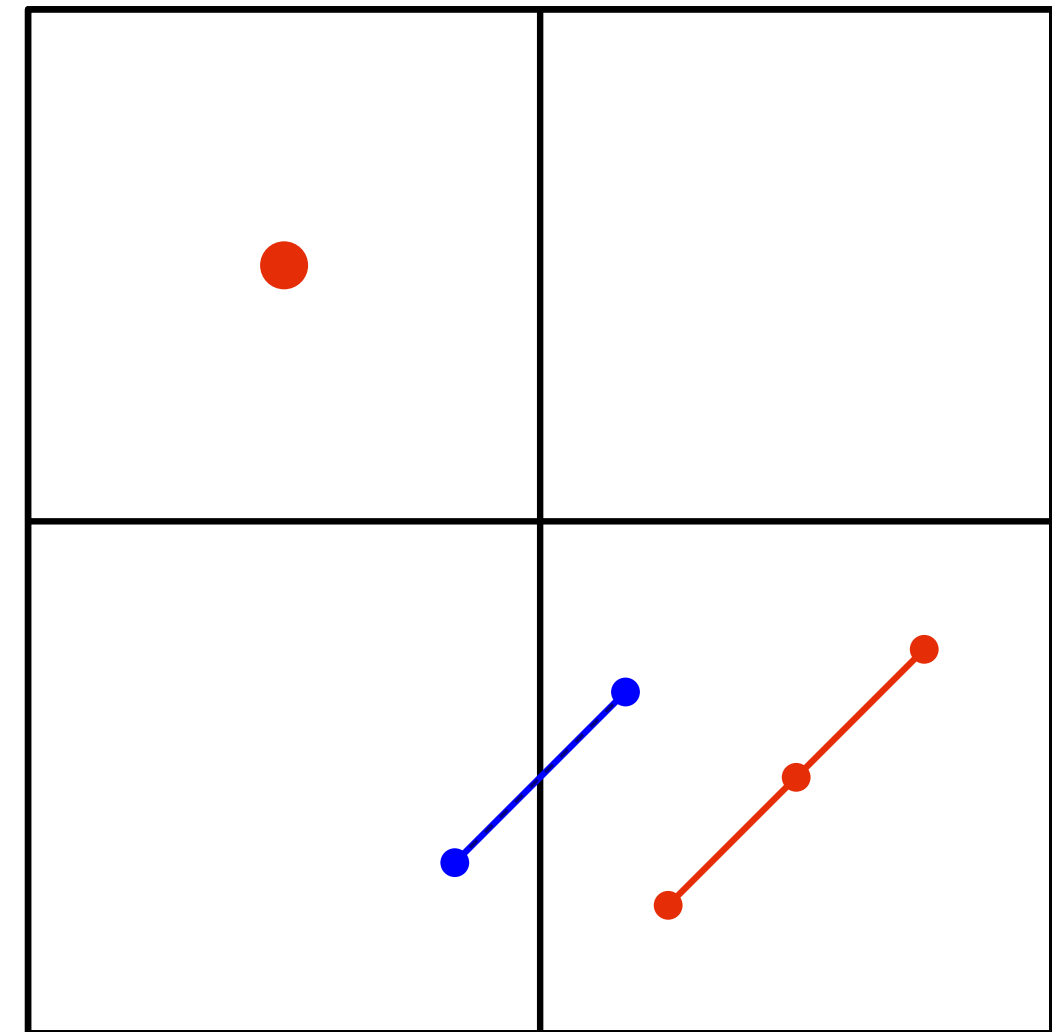
Introduction to Tilings

Definitions:

A **tiling** defines a class of gridded permutations on a fixed grid using *obstructions* and *requirements*:

obstructions are gridded permutations that must be *avoided* by every member of the class

requirements are gridded permutations that must be *contained* by every member of the class



A tiling with obstructions and requirements.

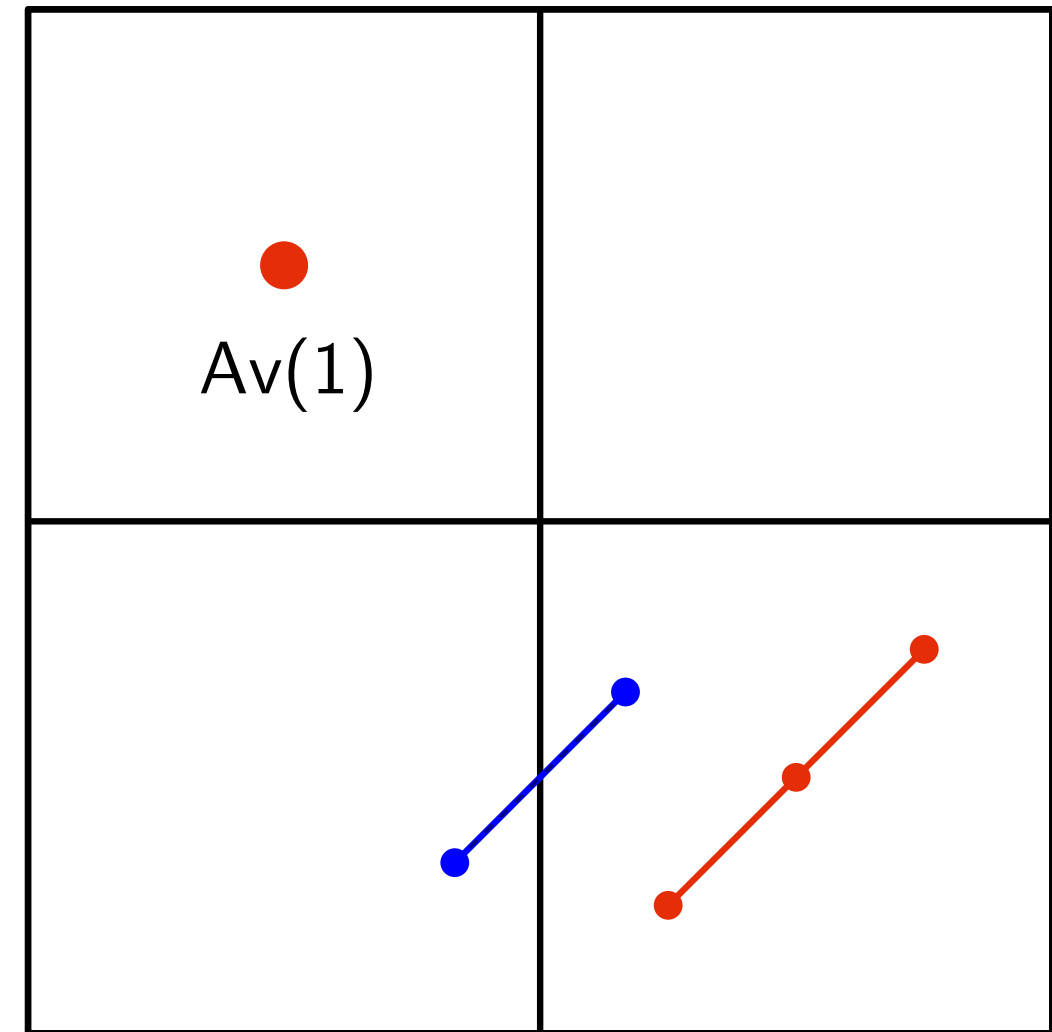
Introduction to Tilings

Definitions:

A **tiling** defines a class of gridded permutations on a fixed grid using *obstructions* and *requirements*:

obstructions are gridded permutations that must be *avoided* by every member of the class

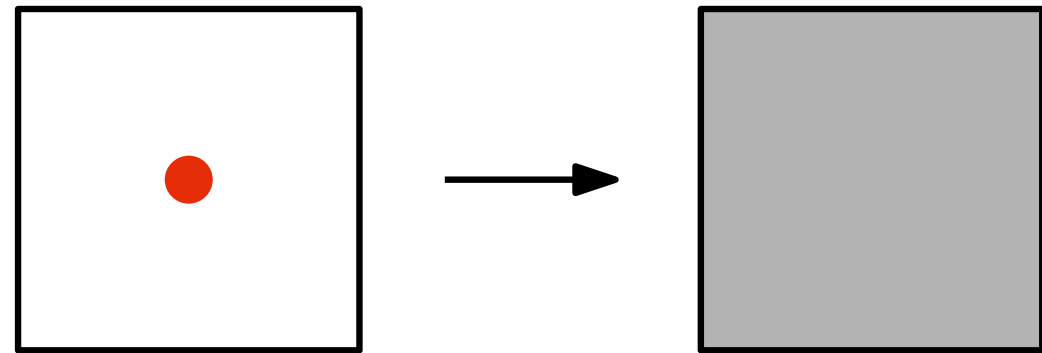
requirements are gridded permutations that must be *contained* by every member of the class



A tiling with obstructions and requirements.

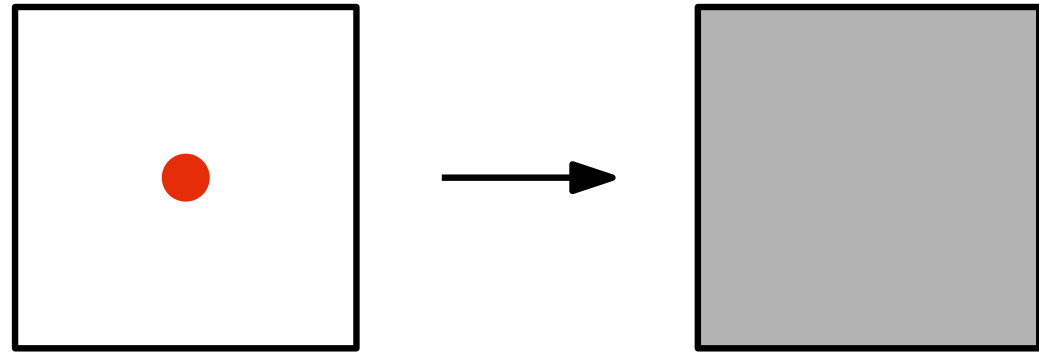
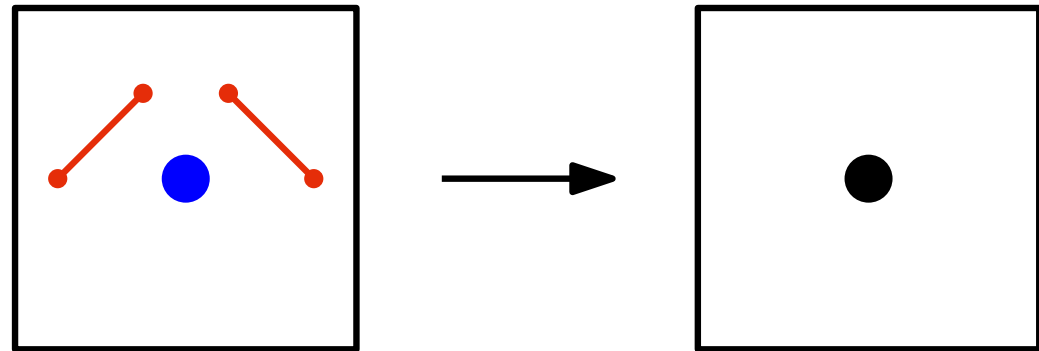


Tiling Shorthands

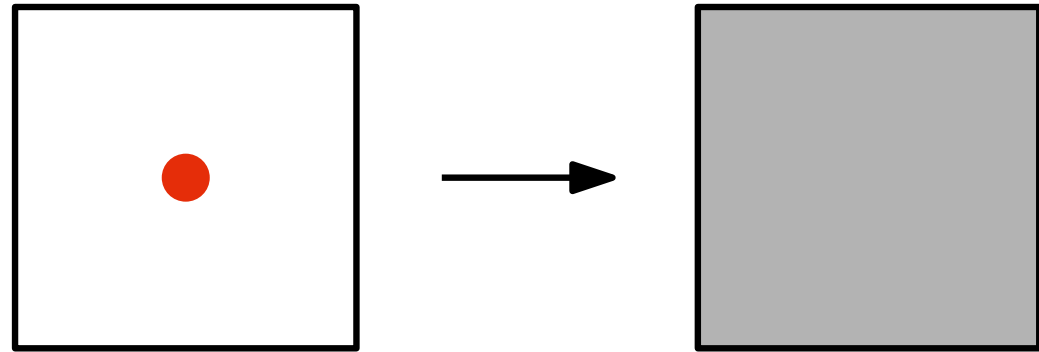
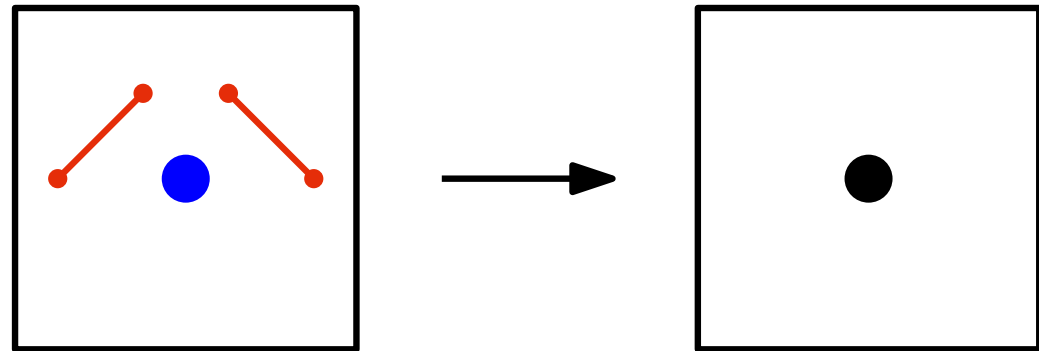
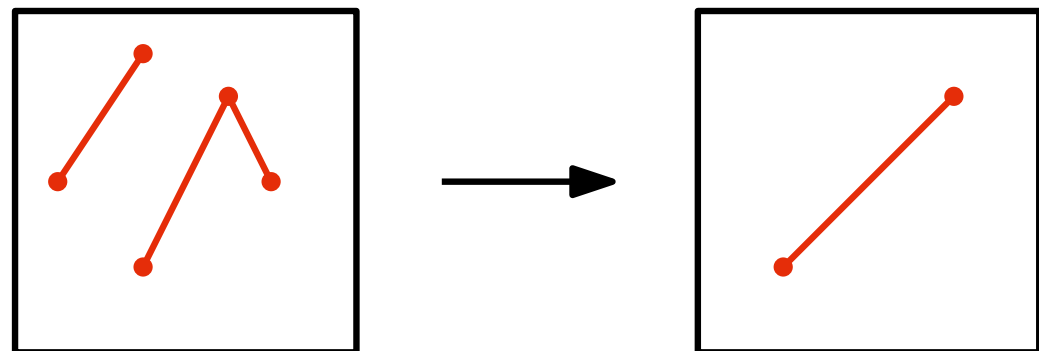


Tiling Shorthands

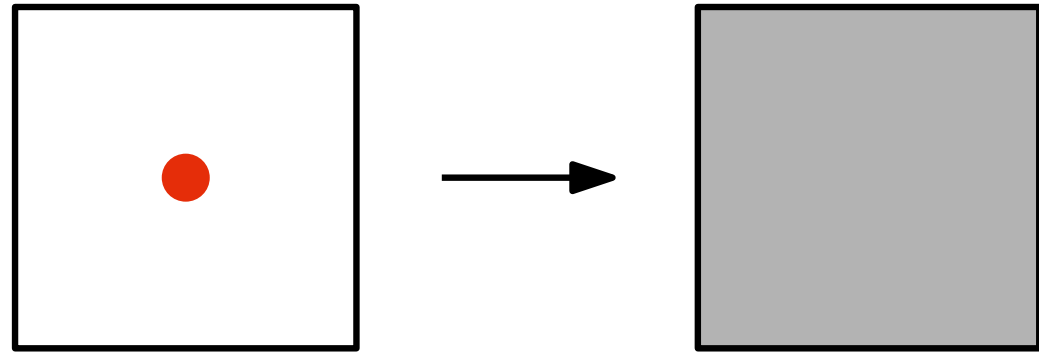
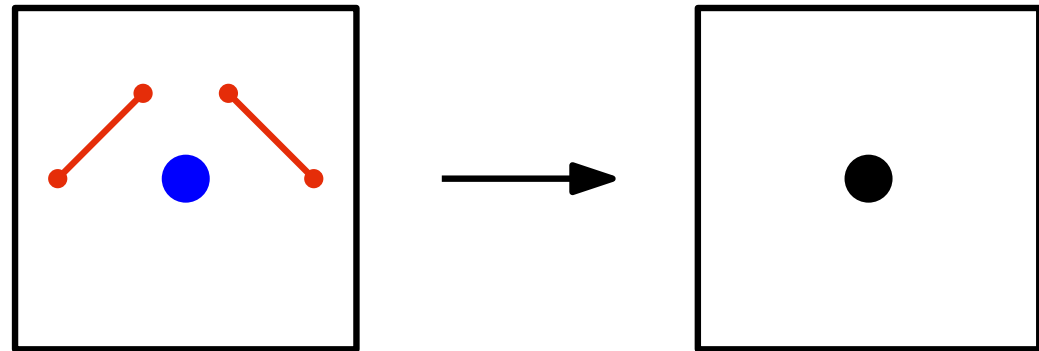
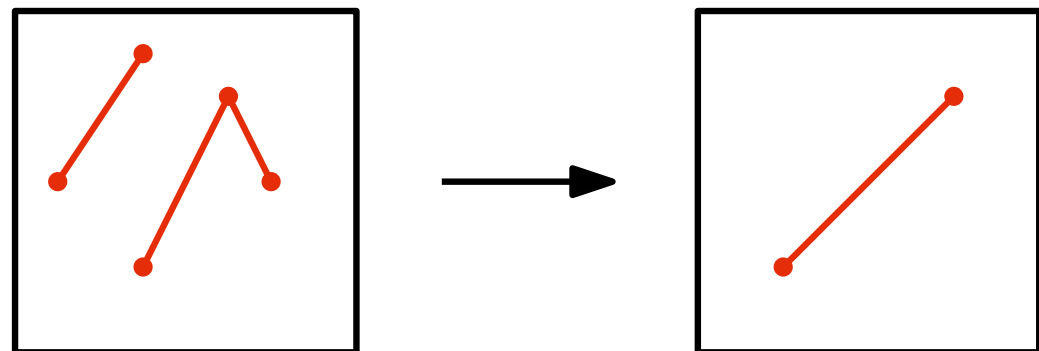
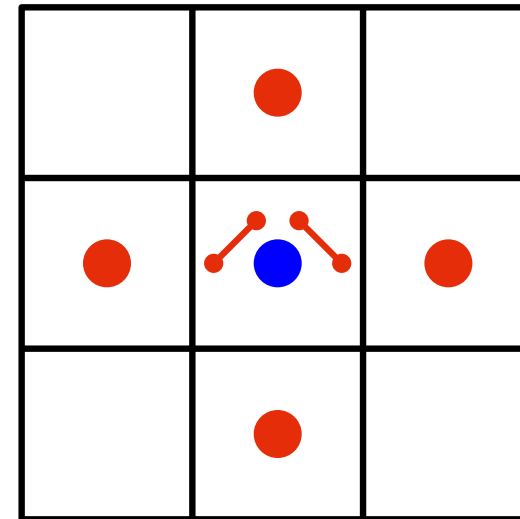
Tiling Shorthands



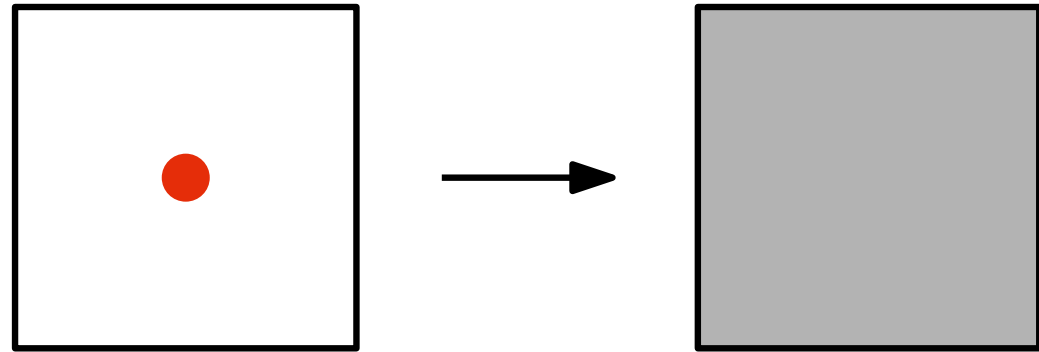
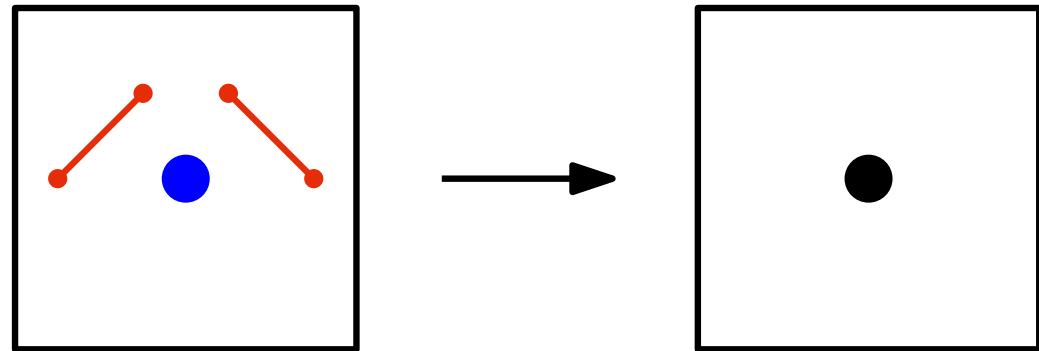
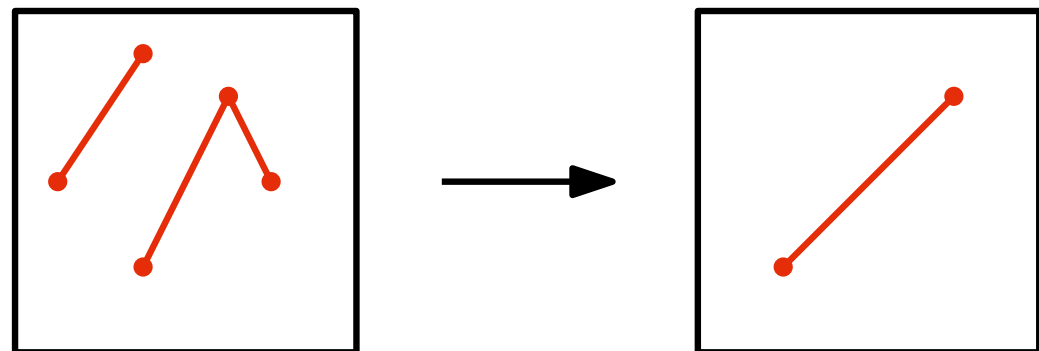
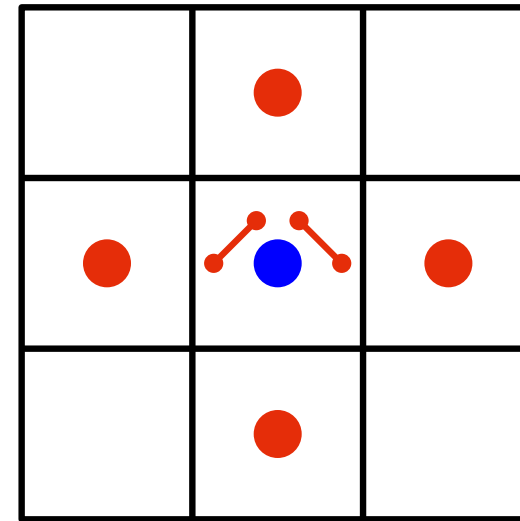
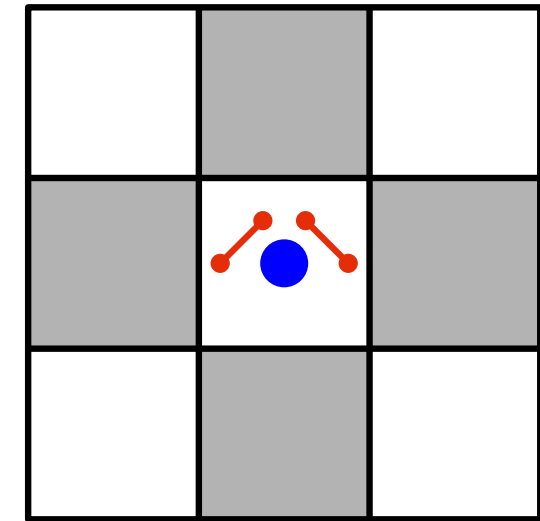
Tiling Shorthands



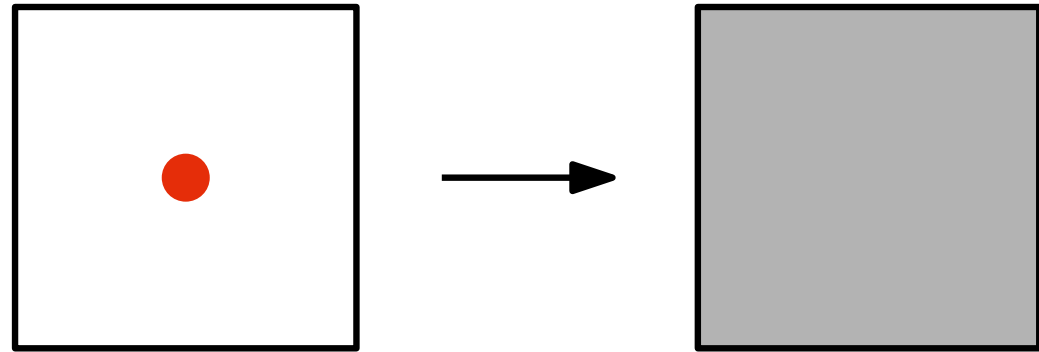
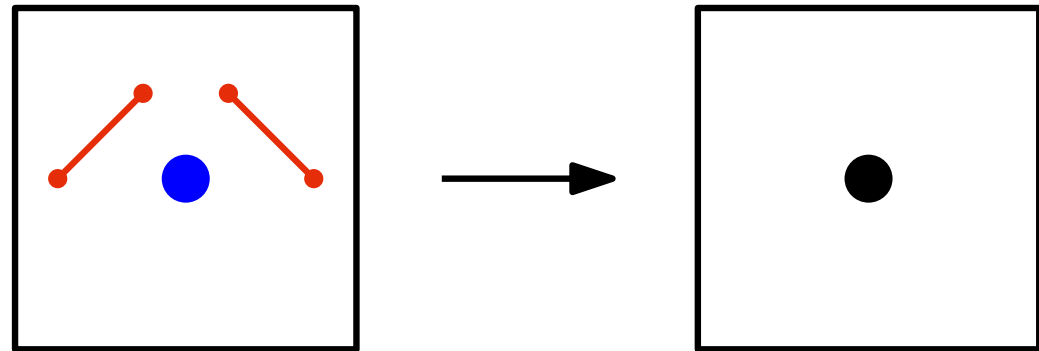
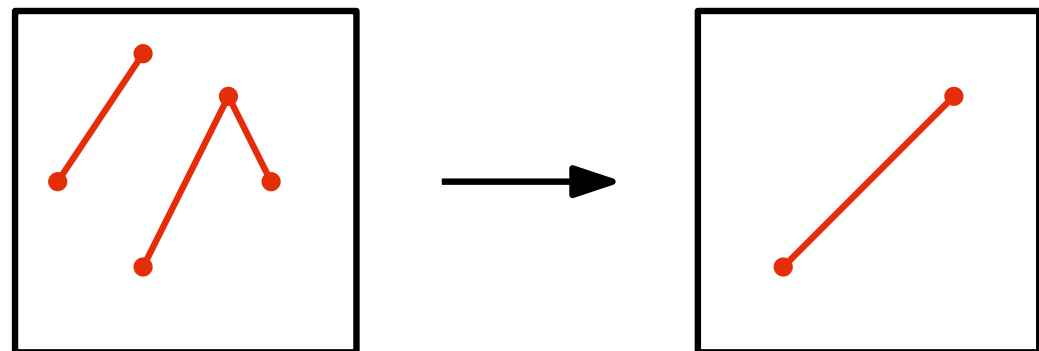
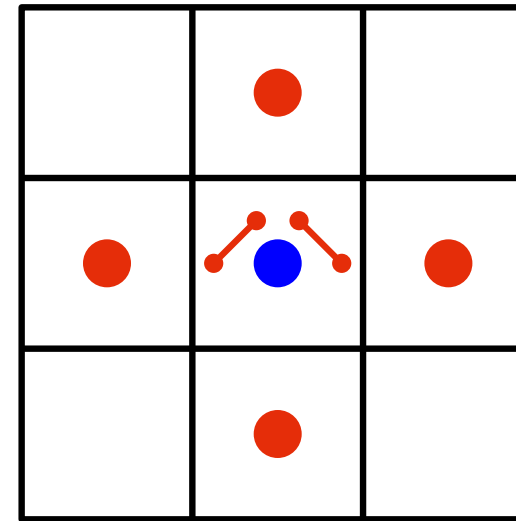
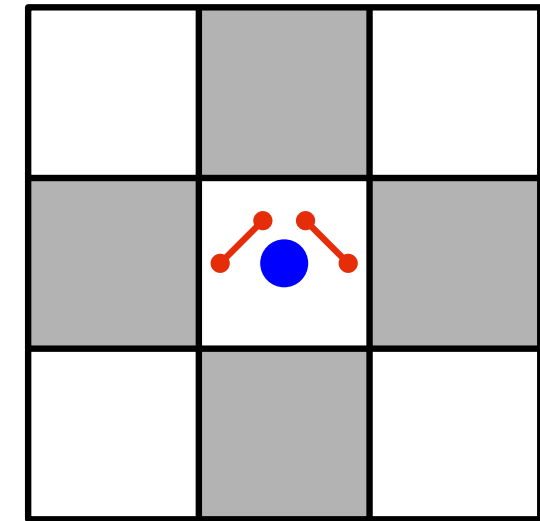
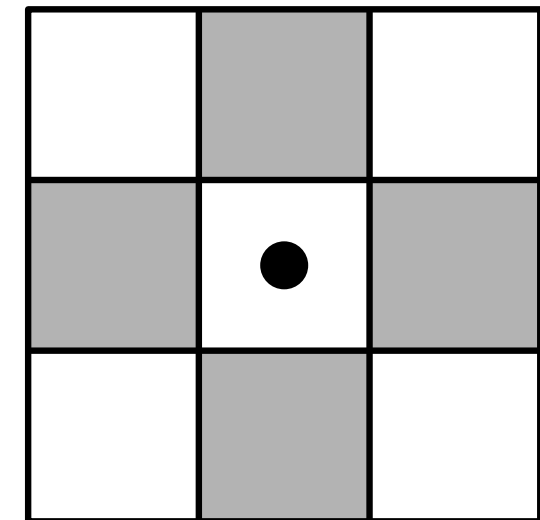
Tiling Shorthands



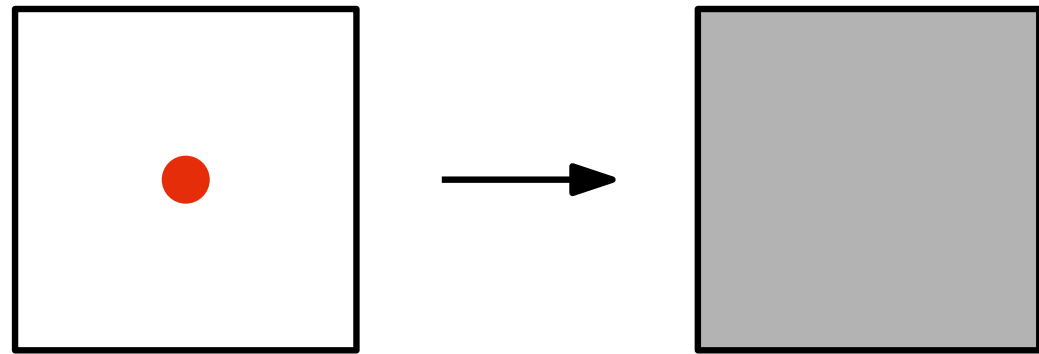
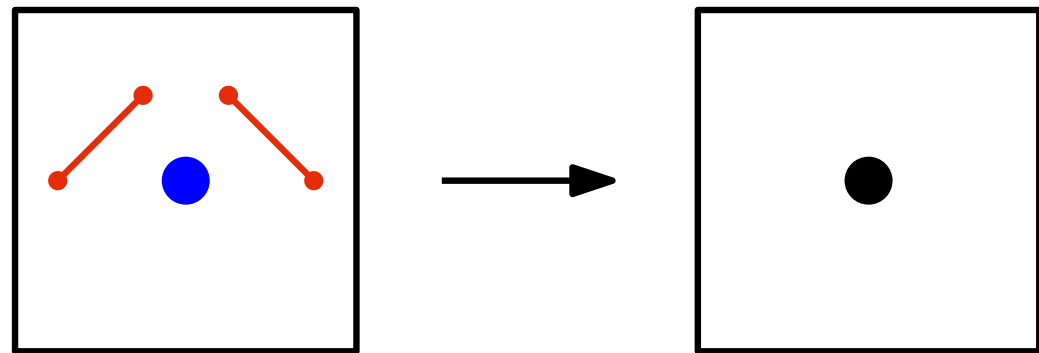
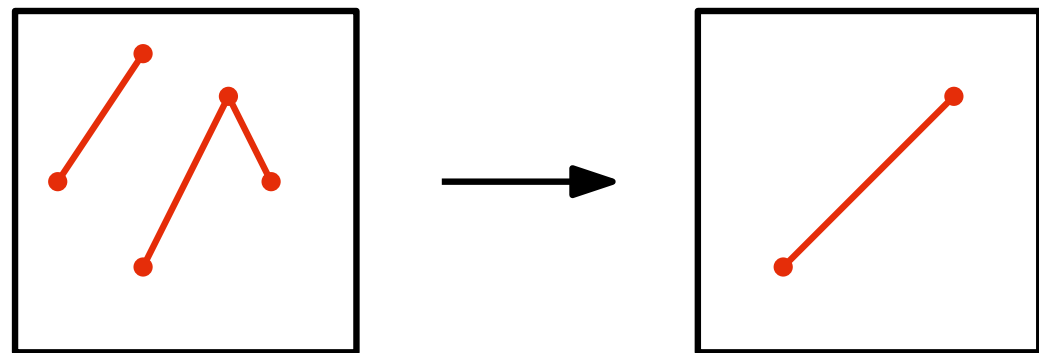
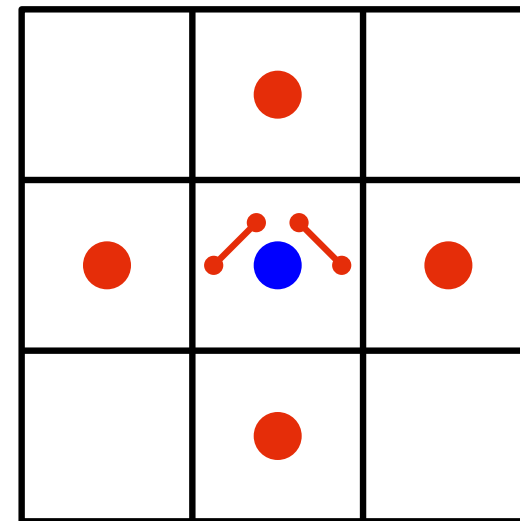
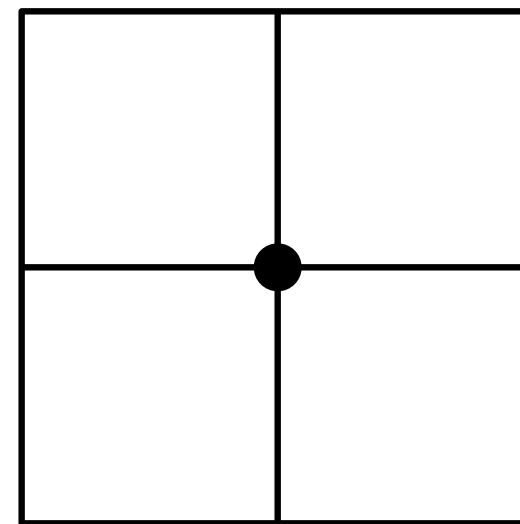
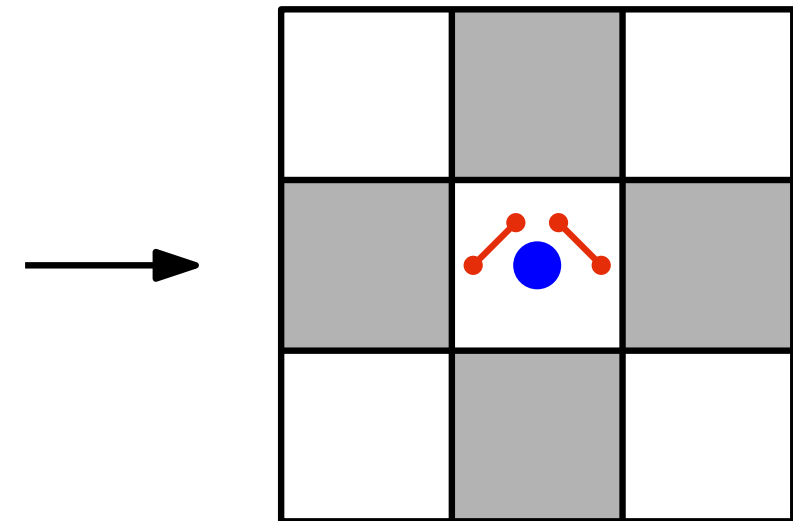
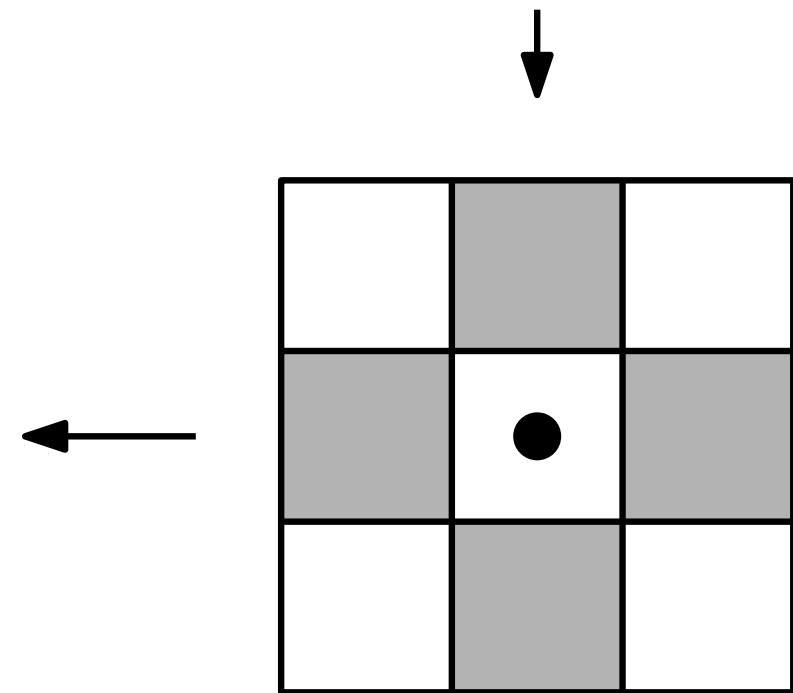
Tiling Shorthands



Tiling Shorthands



Tiling Shorthands





Combinatorial Exploration

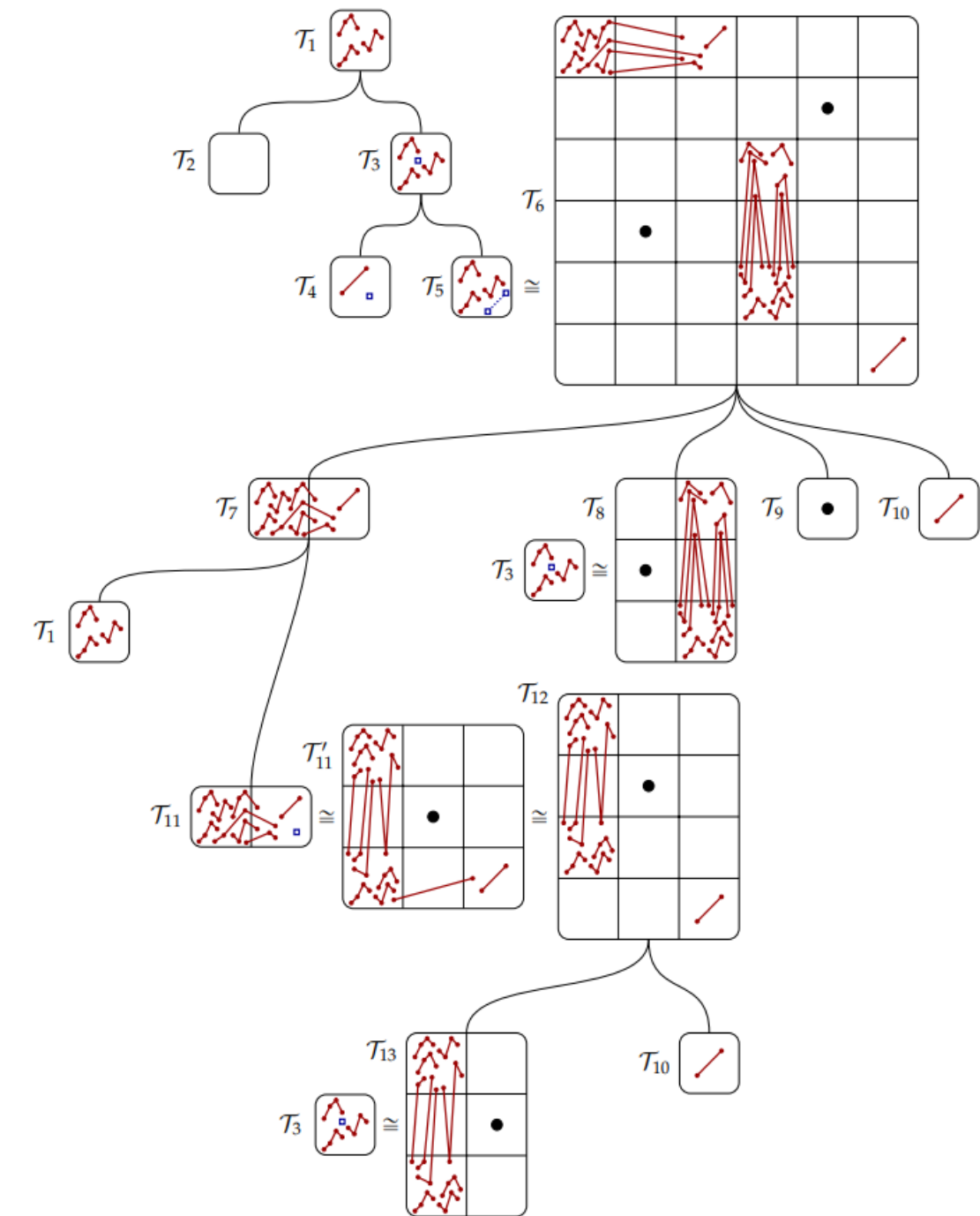


Combinatorial Exploration

- **Tilescope** implements tilings into the Combinatorial Exploration (CombEx) framework. [Albert, Bean, Claeson, Nadeau, Pantone, Ulfarsson]

- **Tilescope** implements tilings into the Combinatorial Exploration (CombEx) framework. [Albert, Bean, Claeson, Nadeau, Pantone, Ulfarsson]
- Starting from a root object, CombEx creates a “universe” of rules between objects from which it extracts a specification and, ideally, a generating function.

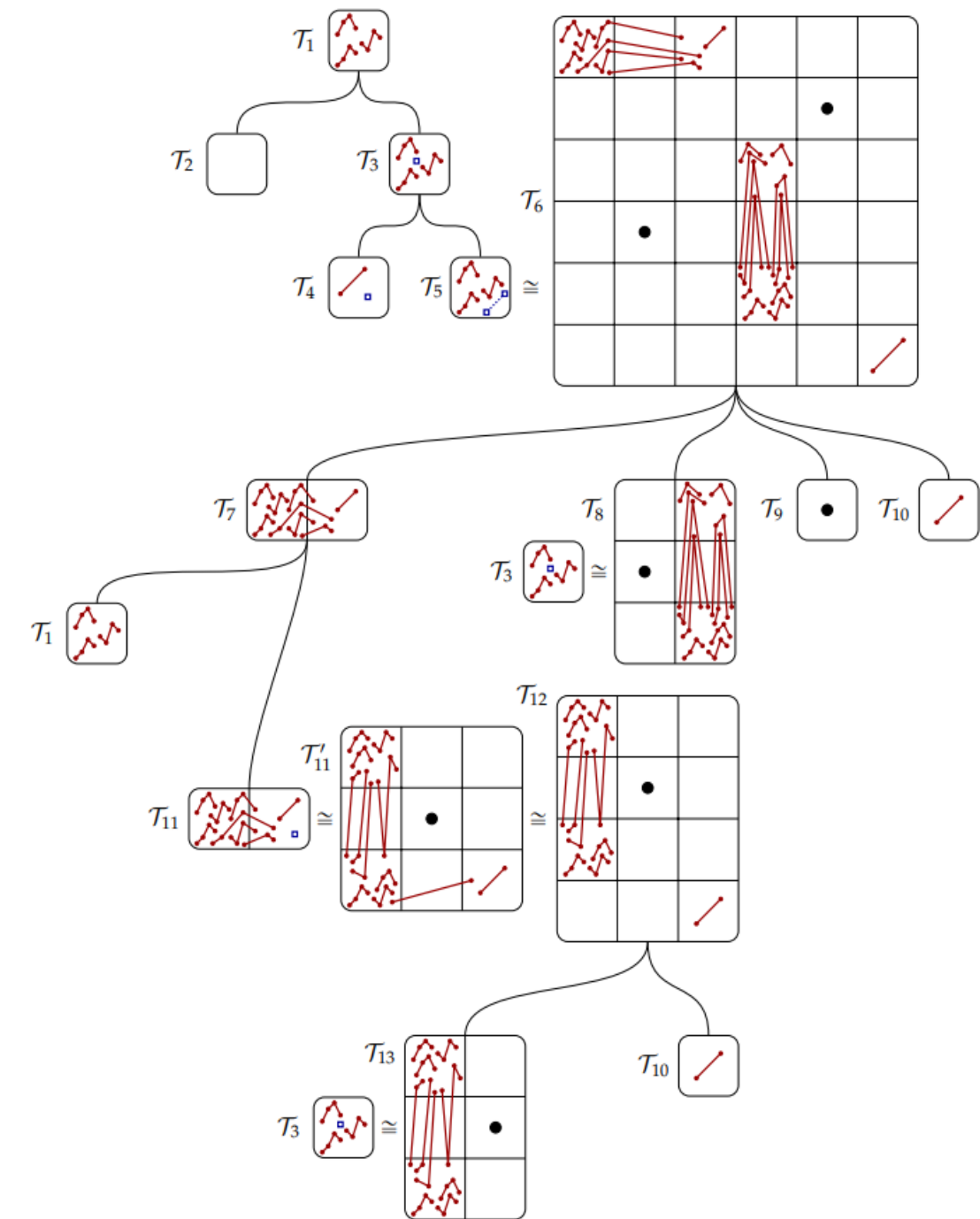
Combinatorial Exploration



The specification for $Av(1243, 1342, 2143)$ found by Combinatorial Exploration.

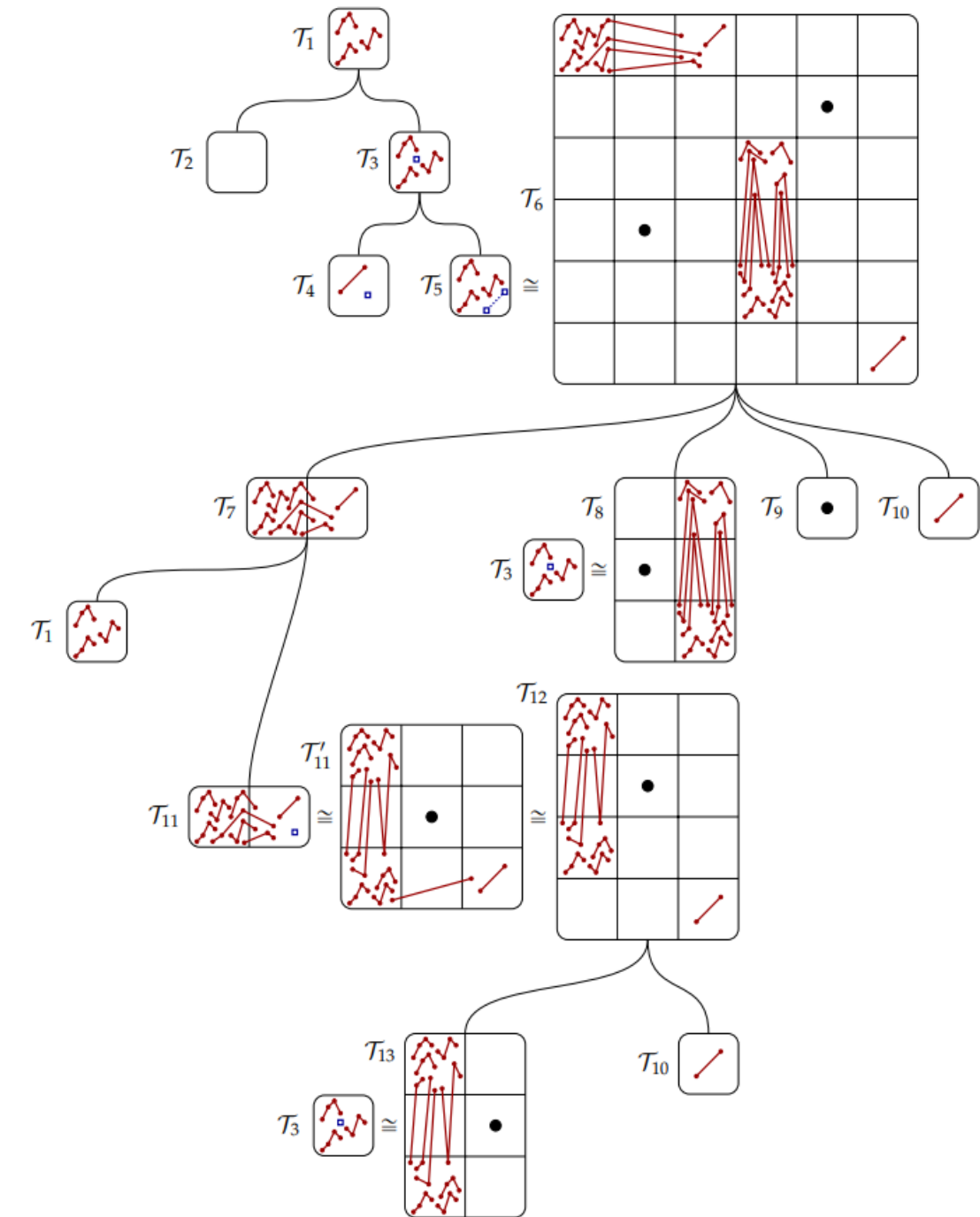
- **Tilescope** implements tilings into the Combinatorial Exploration (CombEx) framework. [Albert, Bean, Claeson, Nadeau, Pantone, Ulfarsson]
- Starting from a root object, CombEx creates a “universe” of rules between objects from which it extracts a specification and, ideally, a generating function.
- Tilescope was extremely successful in enumerating classical pattern avoidance (see the PermPal database), and we are expanding its capabilities.

Combinatorial Exploration



The specification for Av(1243, 1342, 2143) found by Combinatorial Exploration.

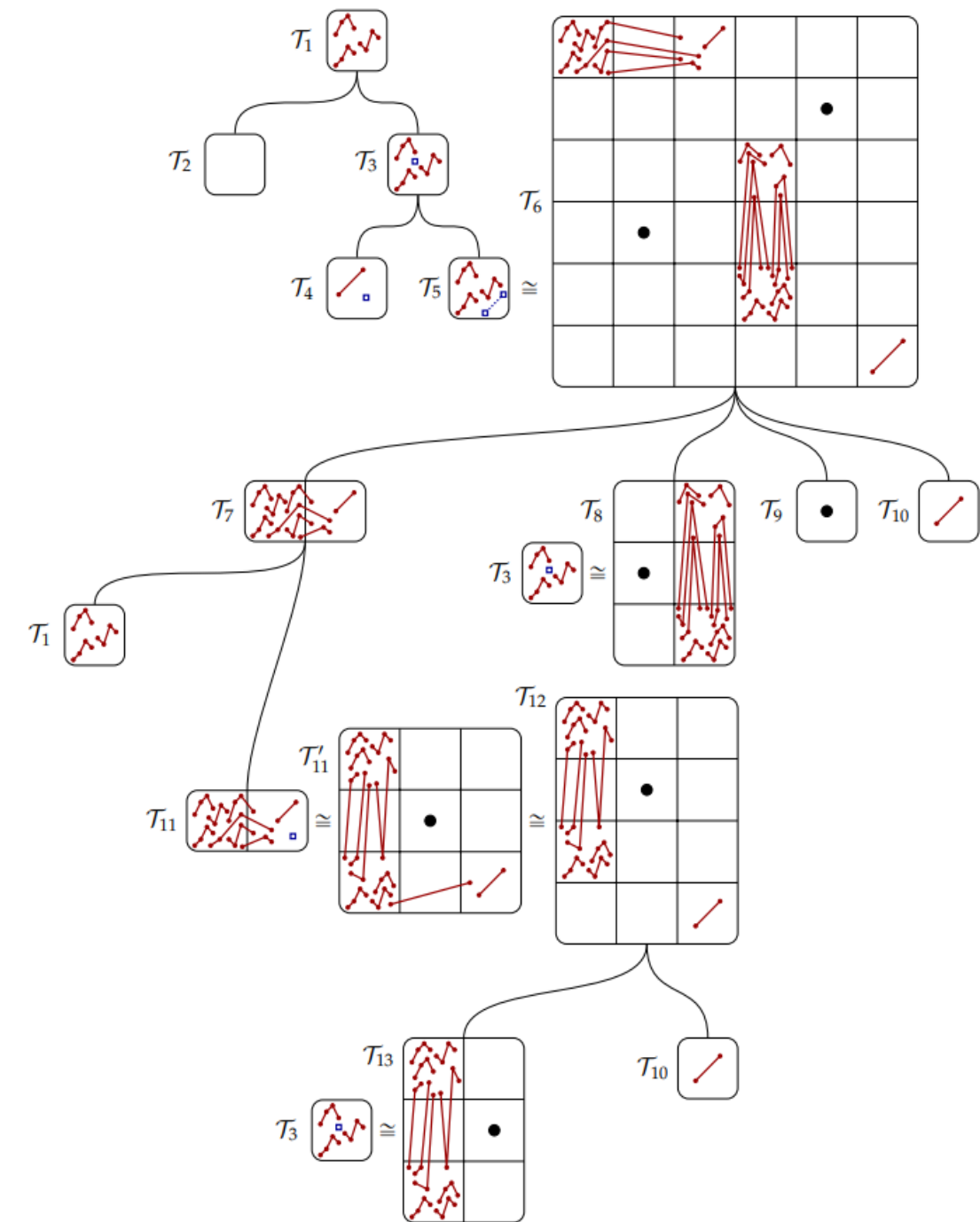
Limitations of Tilings



The specification for $\text{Av}(1243, 1342, 2143)$ found by Combinatorial Exploration.

- Proof trees tend to be quite large and can take a long time to discover.

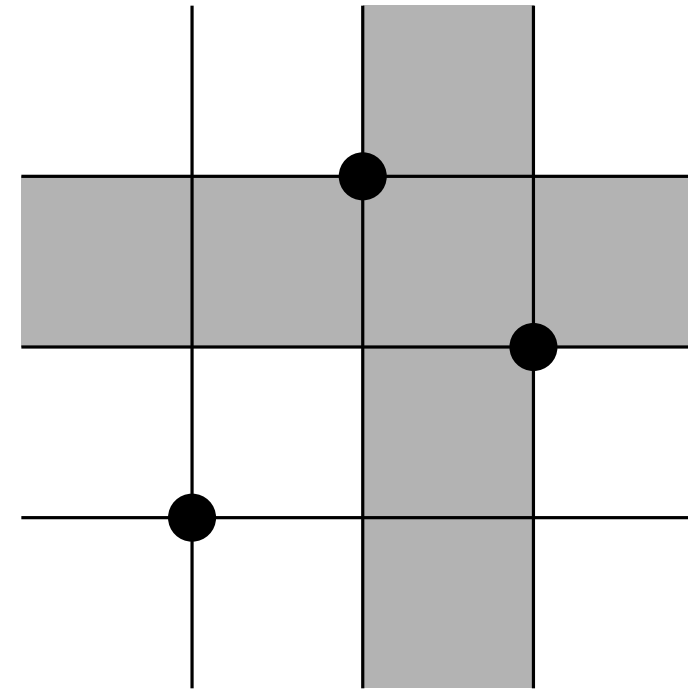
Limitations of Tilings



The specification for $\text{Av}(1243, 1342, 2143)$ found by Combinatorial Exploration.

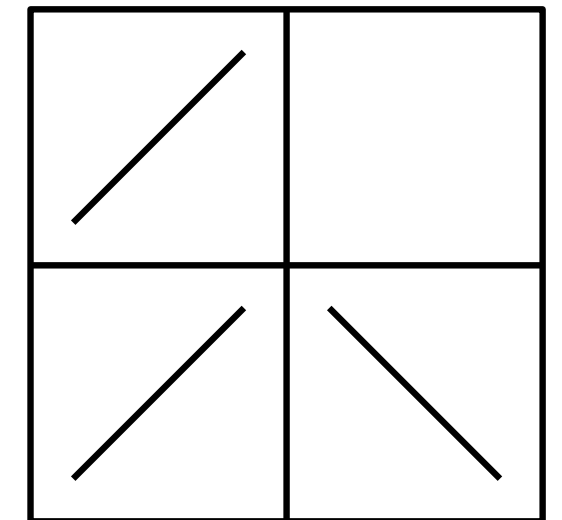
Limitations of Tilings

- Proof trees tend to be quite large and can take a long time to discover.
- Tilings can not be used to represent special kinds of pattern avoidance such as permutation classes defined by mesh patterns or grid classes.



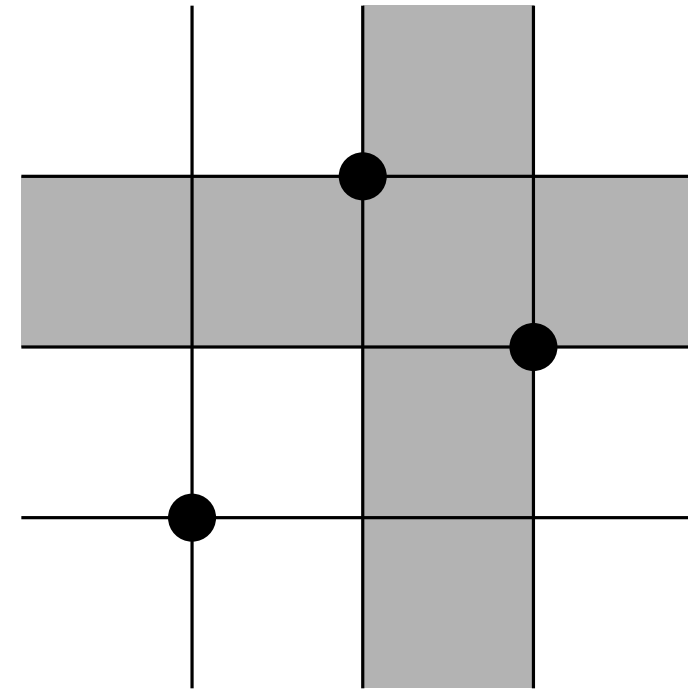
Mesh pattern avoidance is an extension of classical avoidance that requires shaded regions to contain no points.

Grid classes look at all valid griddings of a permutation to find a gridding with a desired “shape.”



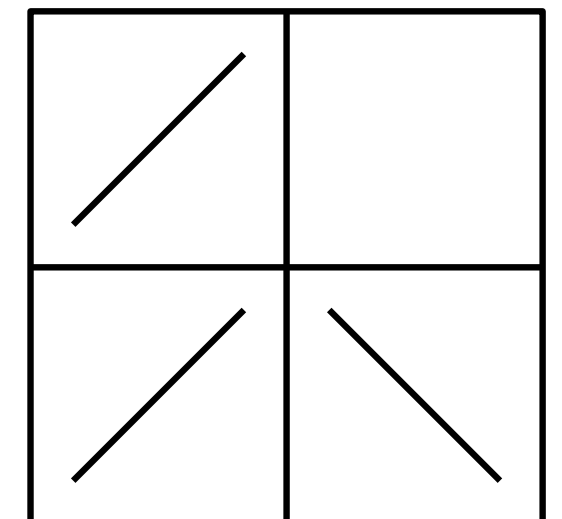
Limitations of Tilings

- Proof trees tend to be quite large and can take a long time to discover.
- Tilings can not be used to represent special kinds of pattern avoidance such as permutation classes defined by mesh patterns or grid classes.
- Refining classes beyond avoidance and containment is impossible.



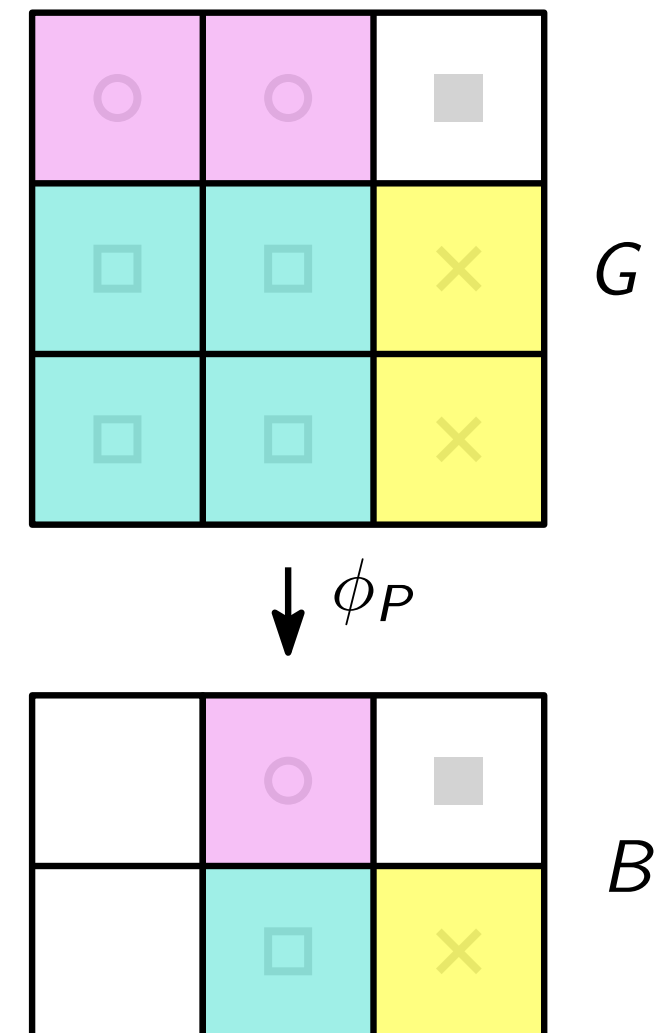
Mesh pattern avoidance is an extension of classical avoidance that requires shaded regions to contain no points.

Grid classes look at all valid griddings of a permutation to find a gridding with a desired “shape.”



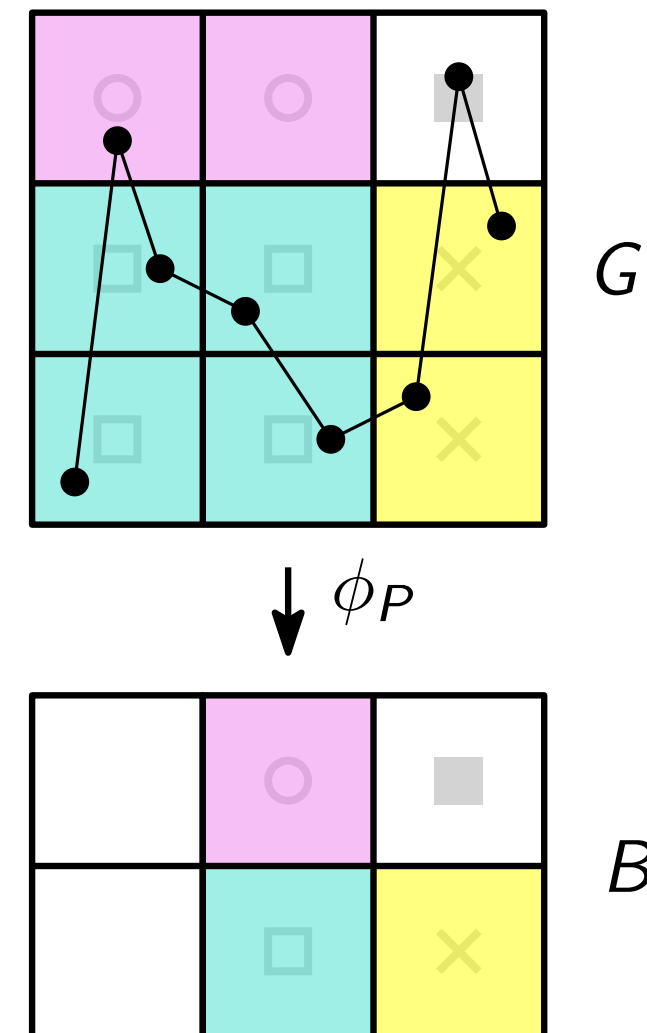
Definition:

A **parameter** P is a tuple (G, m) where G is a tiling we call the “ghost” and m is the row-column map which takes the grid cells of G and maps them to new cells. We use the row-column map to define a function ϕ_P that takes a gridded permutation on G and gives a new gridded permutation with the same underlying pattern.



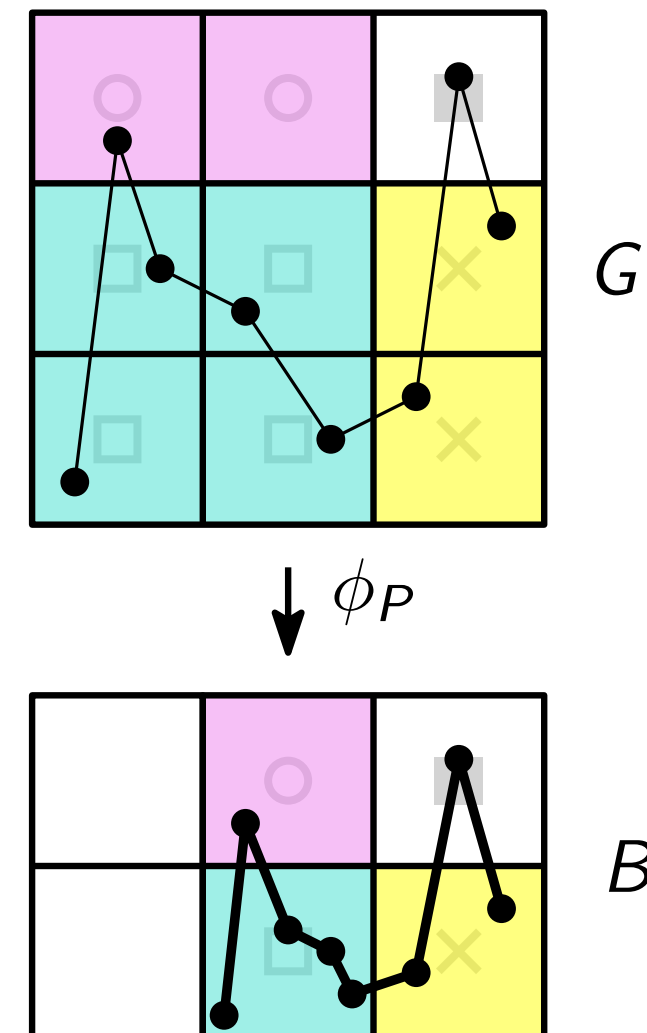
Definition:

A **parameter** P is a tuple (G, m) where G is a tiling we call the “ghost” and m is the row-column map which takes the grid cells of G and maps them to new cells. We use the row-column map to define a function ϕ_P that takes a gridded permutation on G and gives a new gridded permutation with the same underlying pattern.



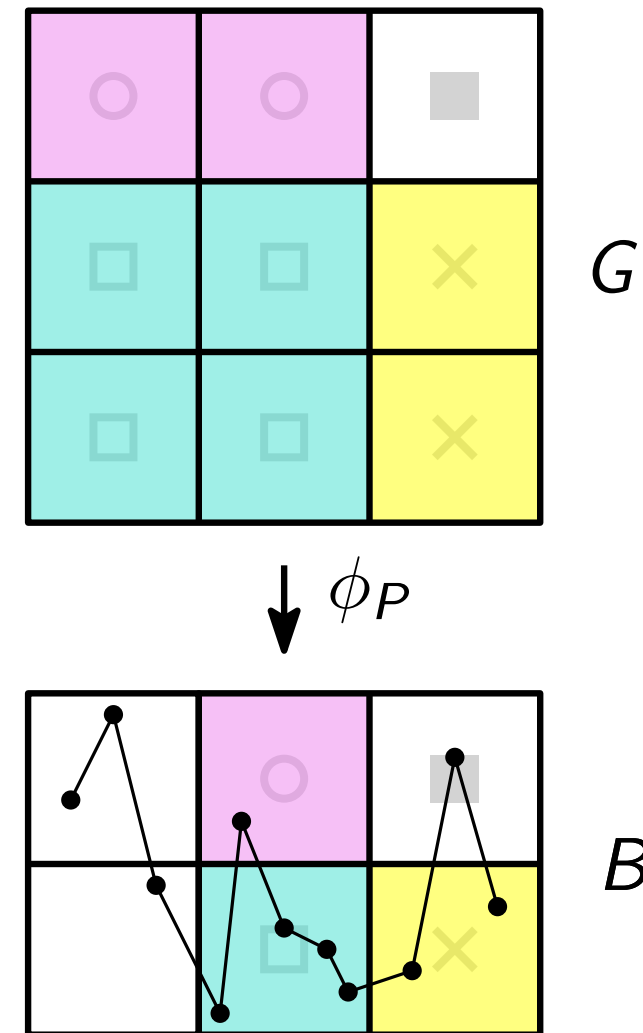
Definition:

A **parameter** P is a tuple (G, m) where G is a tiling we call the “ghost” and m is the row-column map which takes the grid cells of G and maps them to new cells. We use the row-column map to define a function ϕ_P that takes a gridded permutation on G and gives a new gridded permutation with the same underlying pattern.



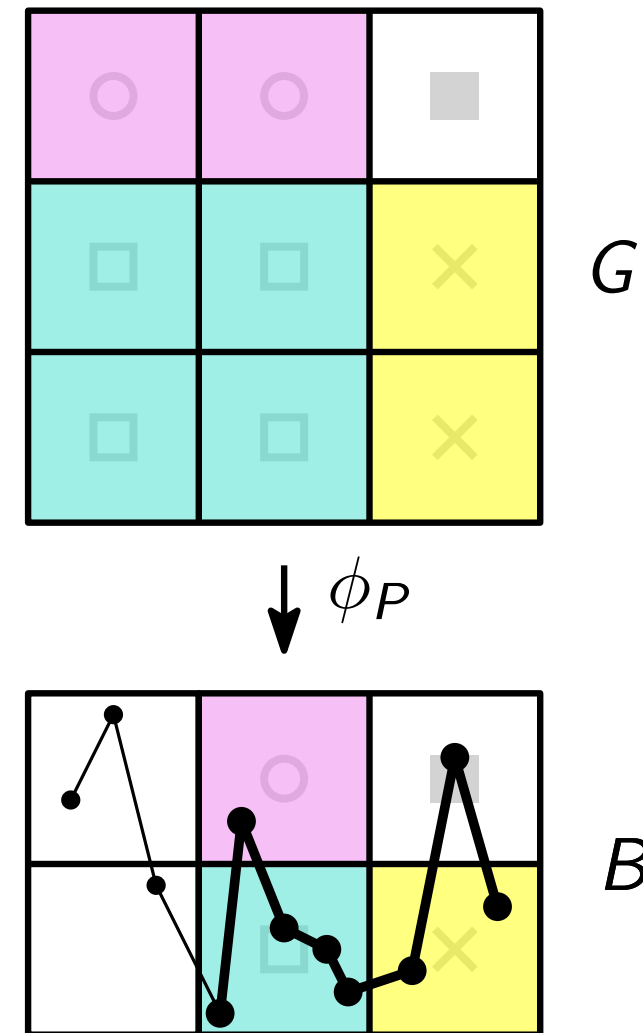
Definition:

A **sub-gridding** of a gridded permutation π is a restriction of π to only include points that lay in specified cells.



Definition:

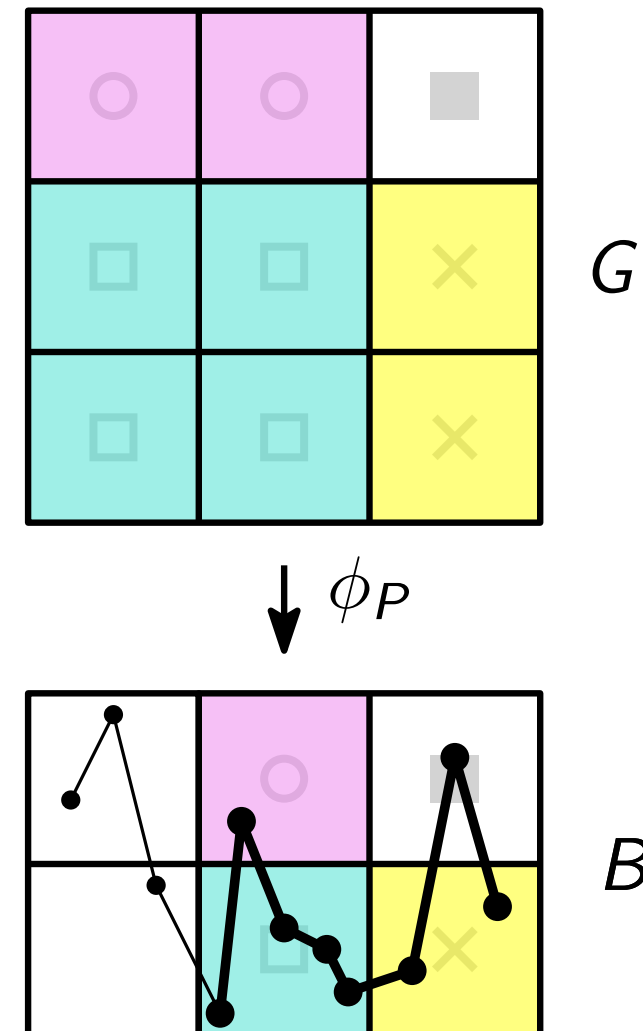
A **sub-gridding** of a gridded permutation π is a restriction of π to only include points that lay in specified cells.



Definition:

A **sub-gridding** of a gridded permutation π is a restriction of π to only include points that lay in specified cells.

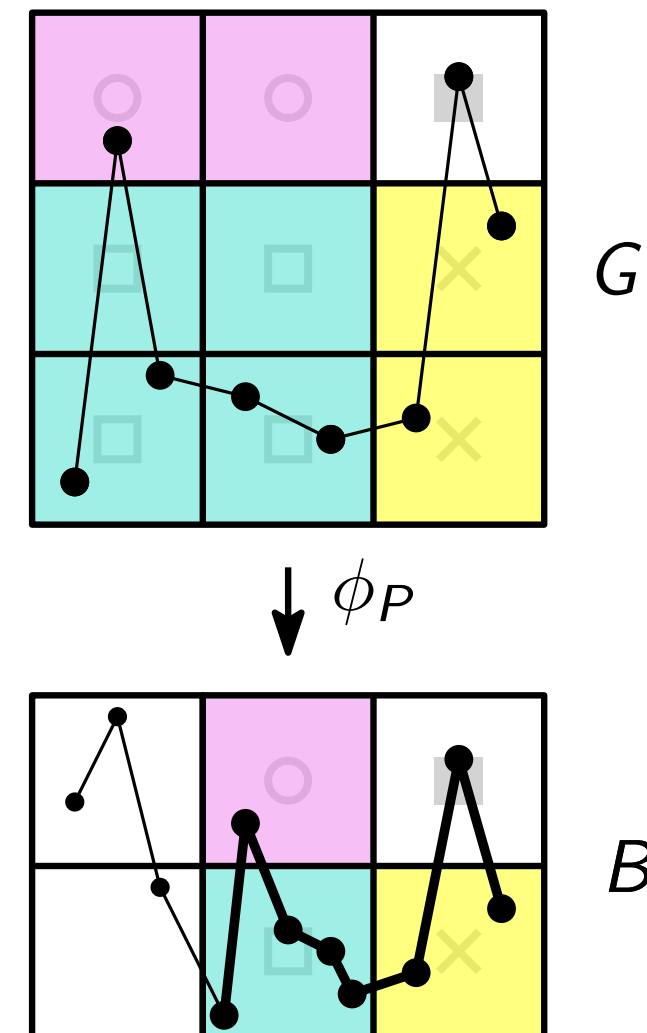
A grided permutation π satisfies a parameter $P = (G, m)$ if the subgridding π' of π whose cells are restricted to the image of m has a preimage in $\phi_P^{-1}(\pi')$ that satisfies G .



Definition:

A **sub-gridding** of a gridded permutation π is a restriction of π to only include points that lay in specified cells.

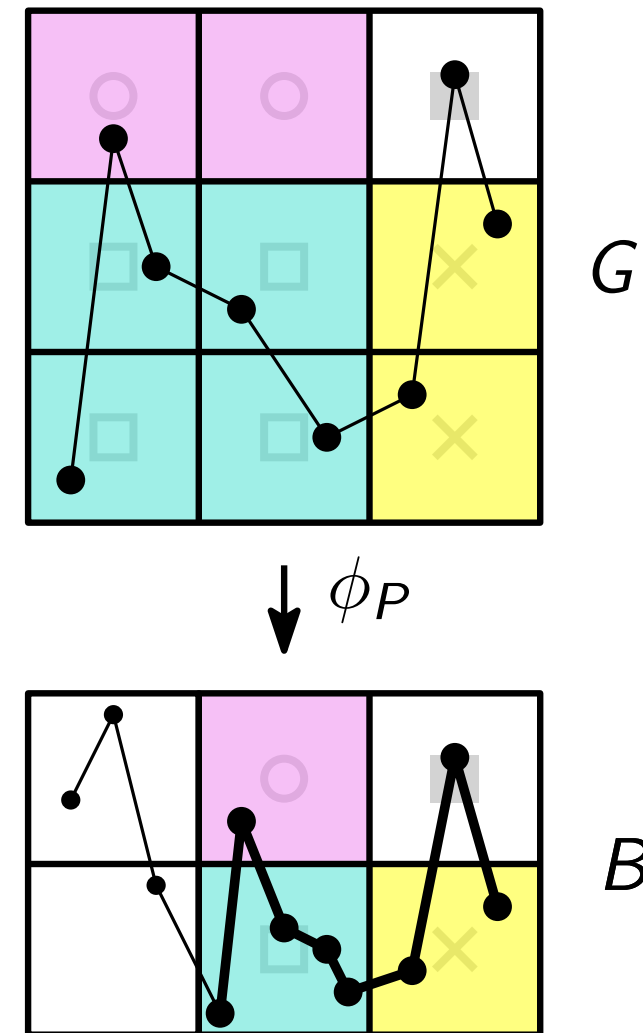
A grided permutation π satisfies a parameter $P = (G, m)$ if the subgridding π' of π whose cells are restricted to the image of m has a preimage in $\phi_P^{-1}(\pi')$ that satisfies G .



Definition:

A **sub-gridding** of a gridded permutation π is a restriction of π to only include points that lay in specified cells.

A grided permutation π satisfies a parameter $P = (G, m)$ if the subgridding π' of π whose cells are restricted to the image of m has a preimage in $\phi_P^{-1}(\pi')$ that satisfies G .

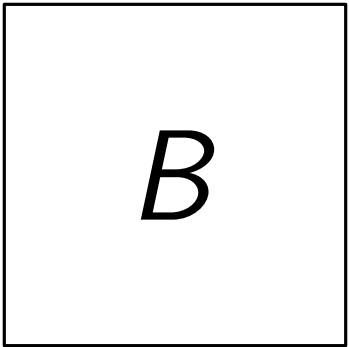


Definition:

A **mapped tiling** (or “mapping”) has a base tiling B with three collections of parameters: avoiders, containers, and enumerators. Each of these collections contain parameters which map to the cells of B and serve a different purpose to define a class of gridded permutations.

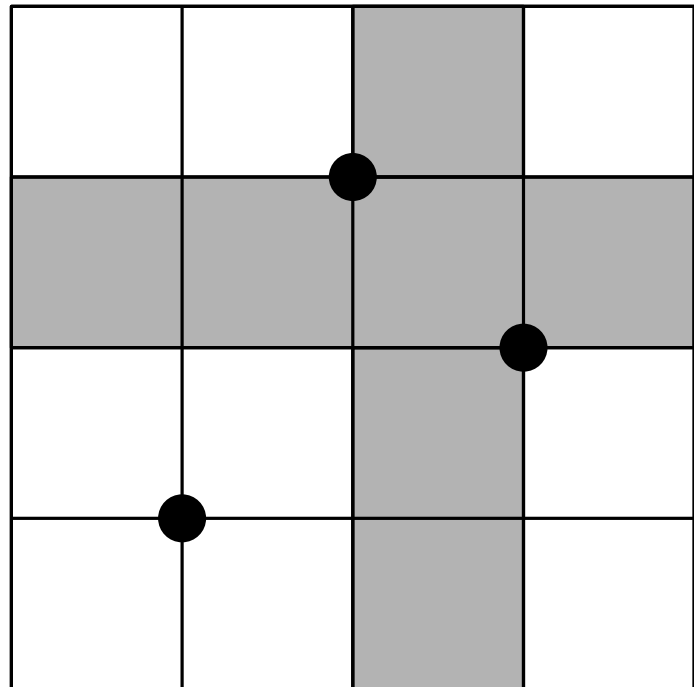
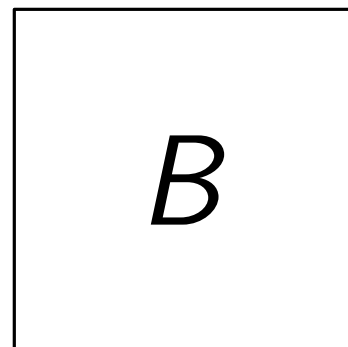


Mapped Tilings



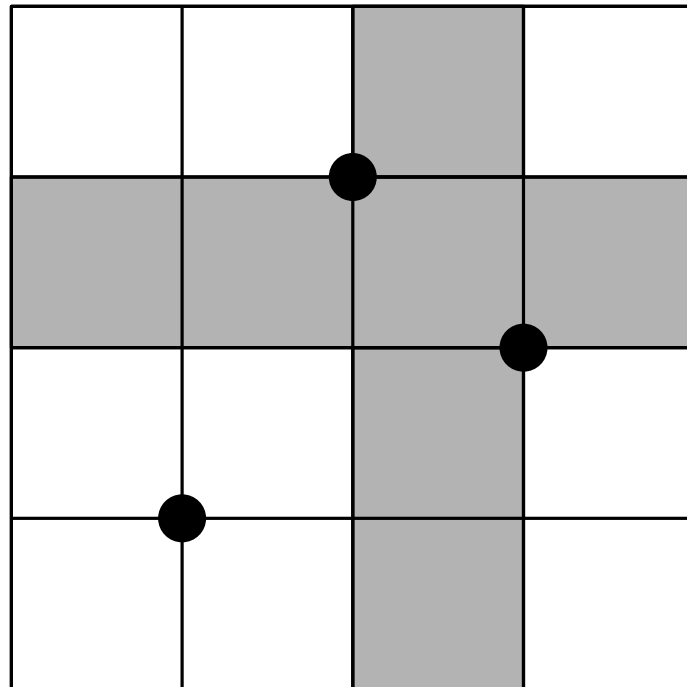
Avoiders

Exclude gridded permutations on B with a valid preimage.



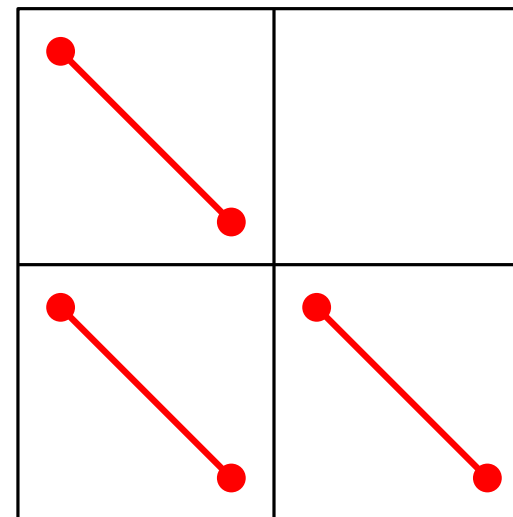
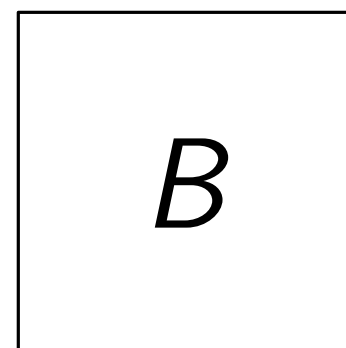
Avoiders

Exclude gridded permutations on B with a valid preimage.



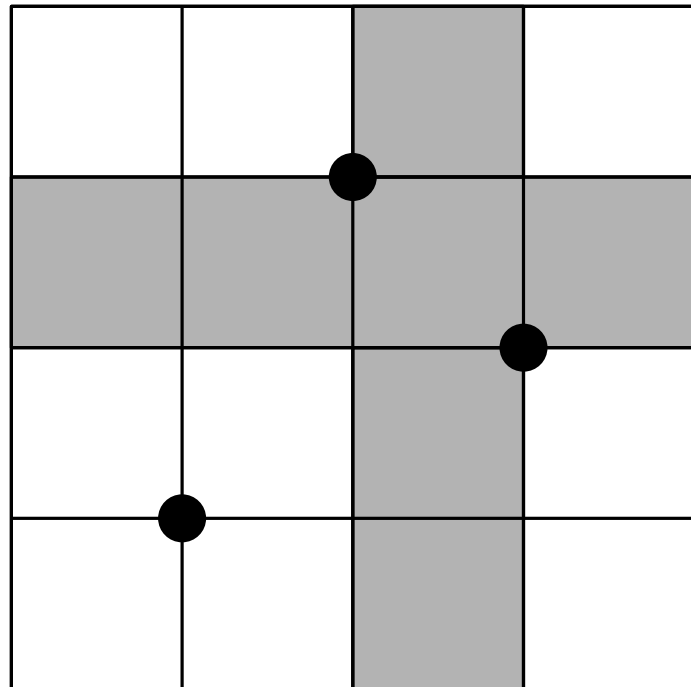
Containers

Only allow gridded permutations on B with a valid preimage.



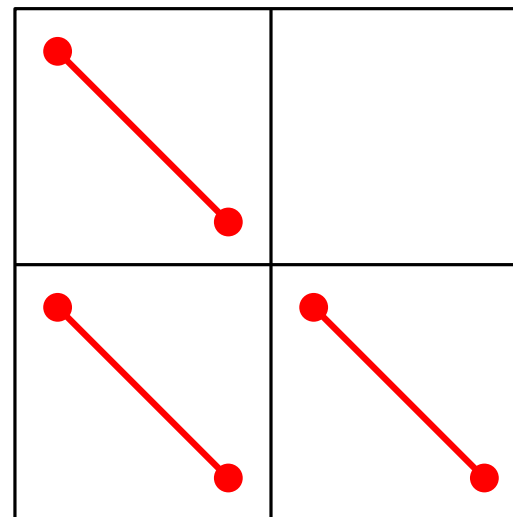
Avoiders

Exclude gridded permutations on B with a valid preimage.



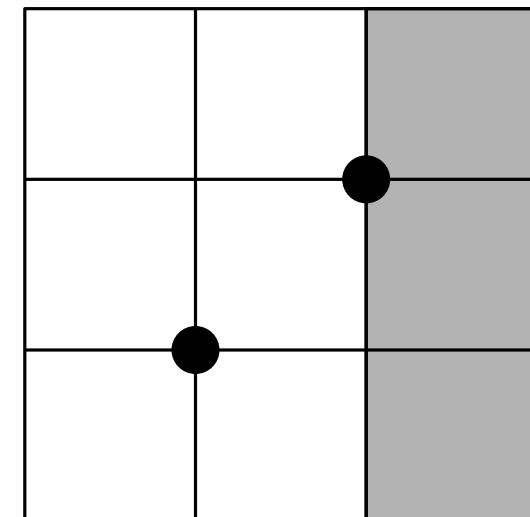
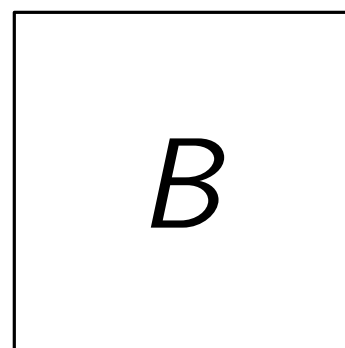
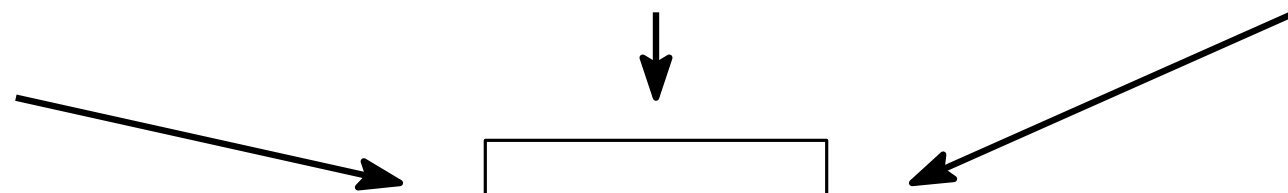
Containers

Only allow gridded permutations on B with a valid preimage.



Enumerators

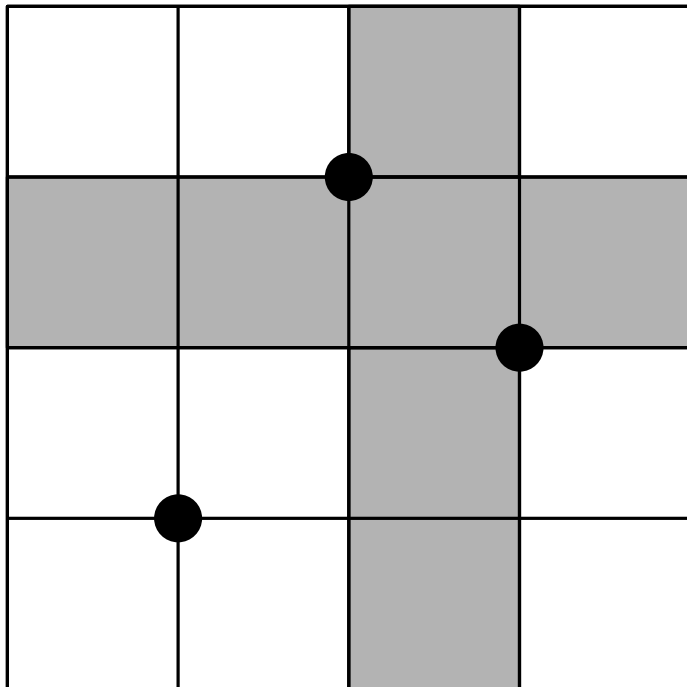
Count the number of valid preimages a gridded permutation on B has.



Mapped Tilings (cont.)

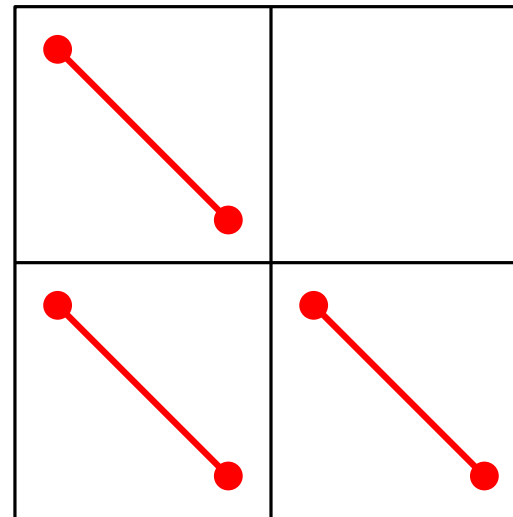
Avoiders

Exclude gridded permutations on B with a valid preimage.



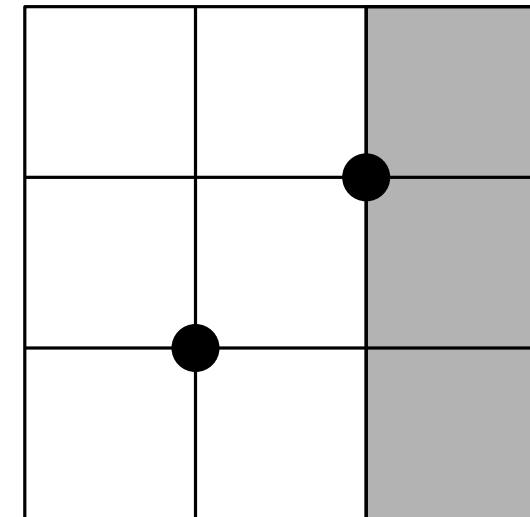
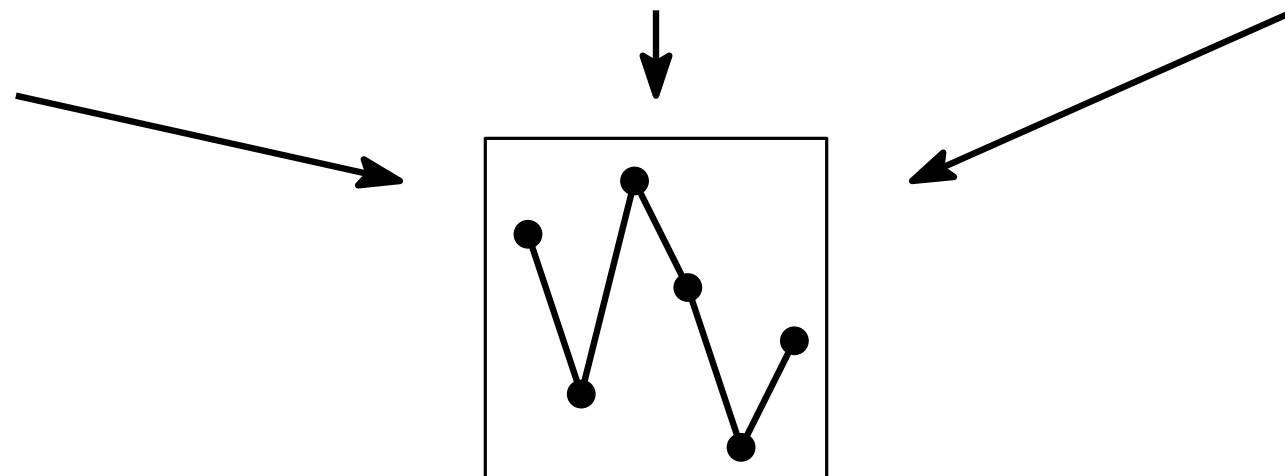
Containers

Only allow gridded permutations on B with a valid preimage.



Enumerators

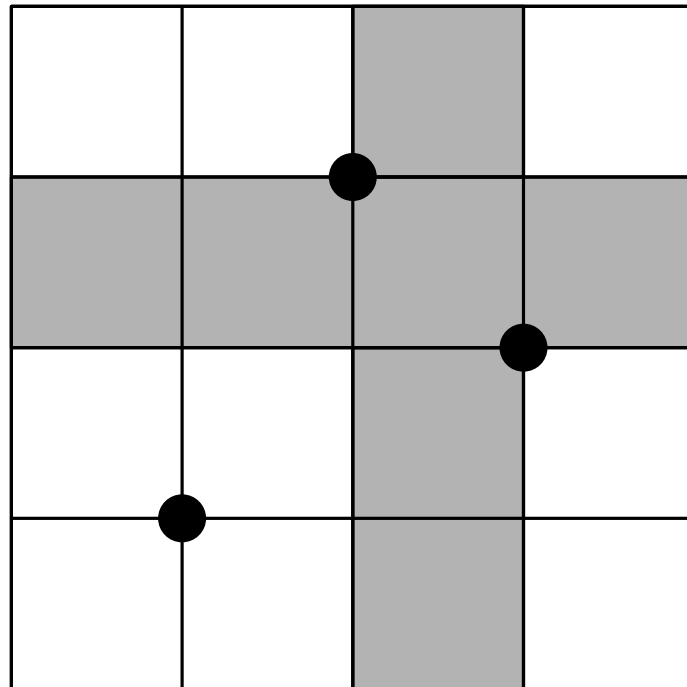
Count the number of valid preimages a gridded permutation on B has.



Mapped Tilings (cont.)

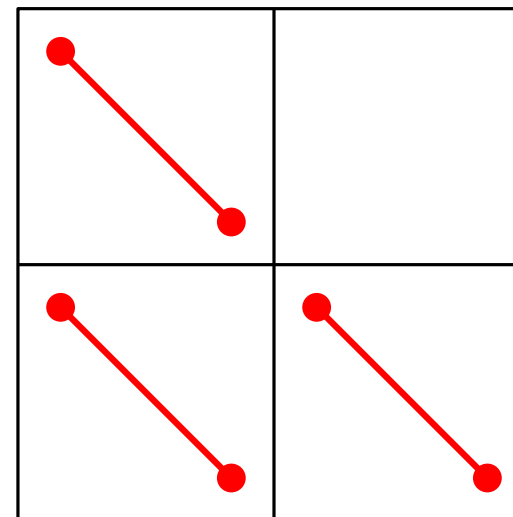
Avoiders

526413 has no valid preimage



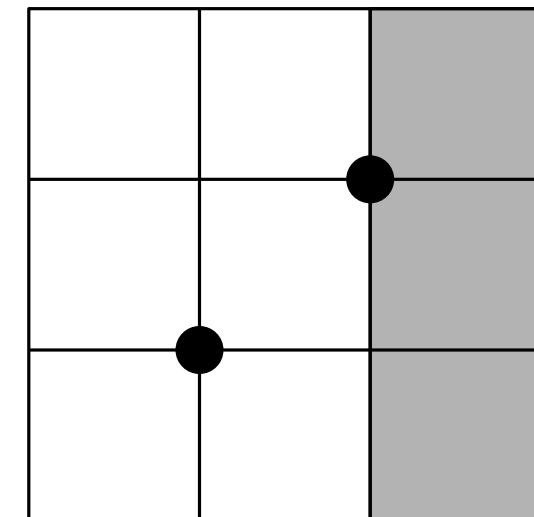
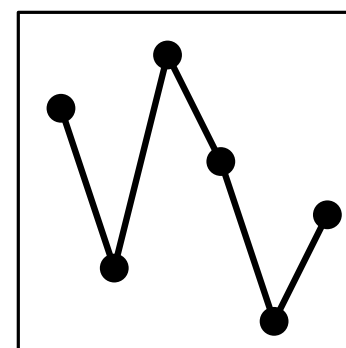
Containers

Only allow gridded permutations on B with a valid preimage.



Enumerators

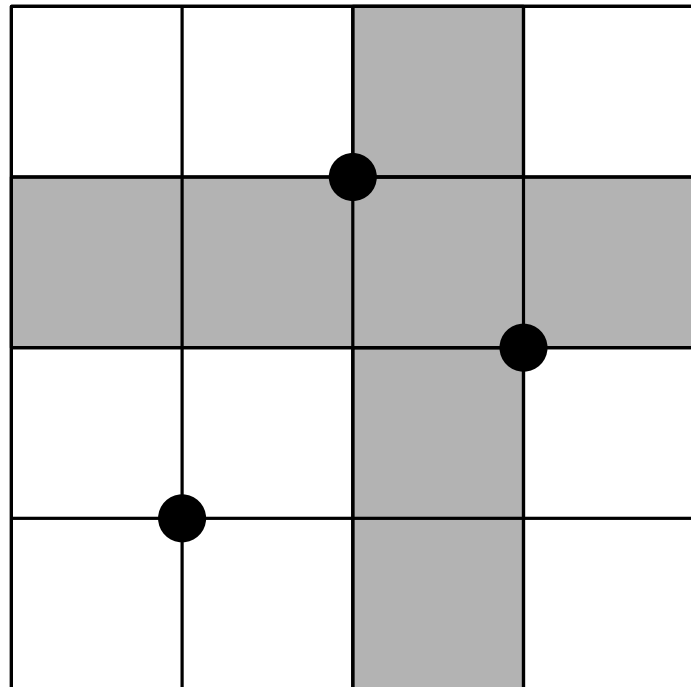
Count the number of valid preimages a gridded permutation on B has.



Mapped Tilings (cont.)

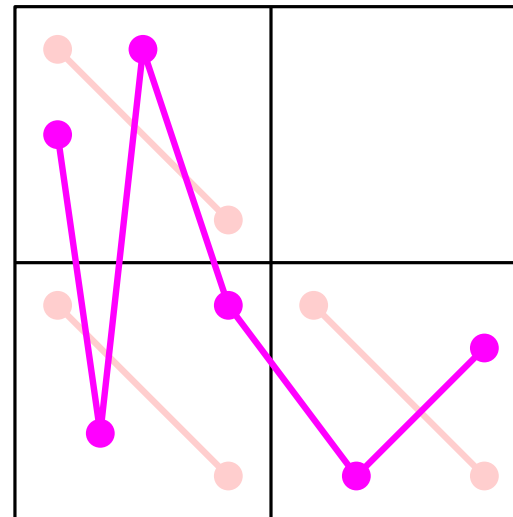
Avoiders

526413 has no valid preimage



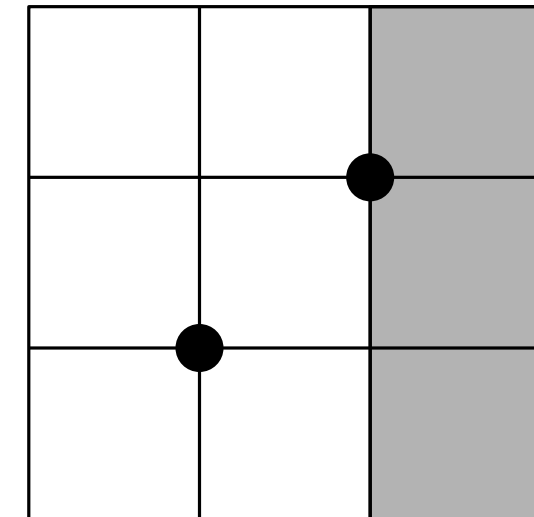
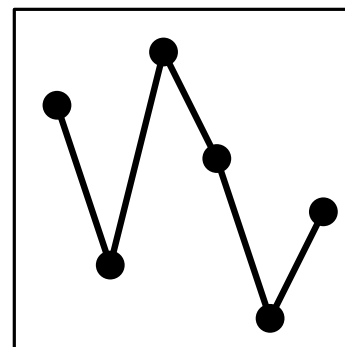
Containers

526413 has a valid preimage.



Enumerators

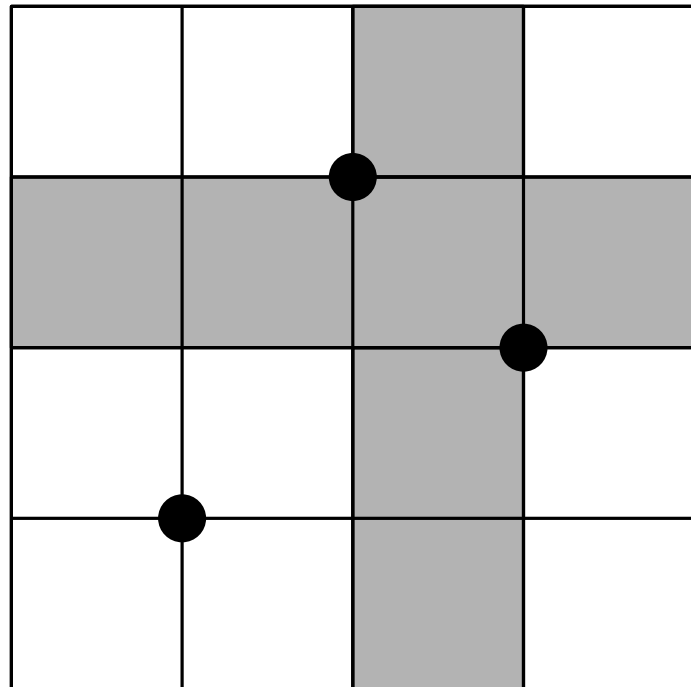
Count the number of valid preimages a gridded permutation on B has.



Mapped Tilings (cont.)

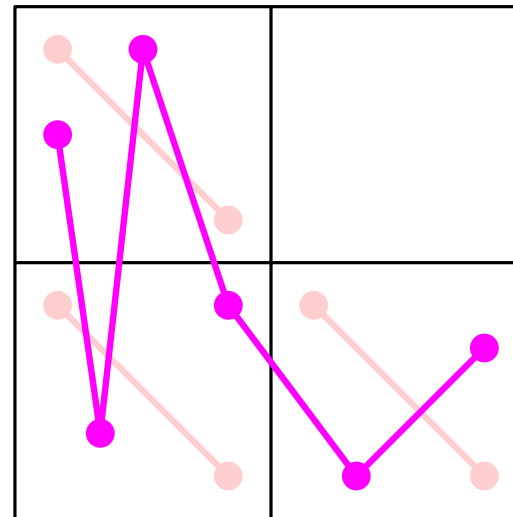
Avoiders

526413 has no valid preimage



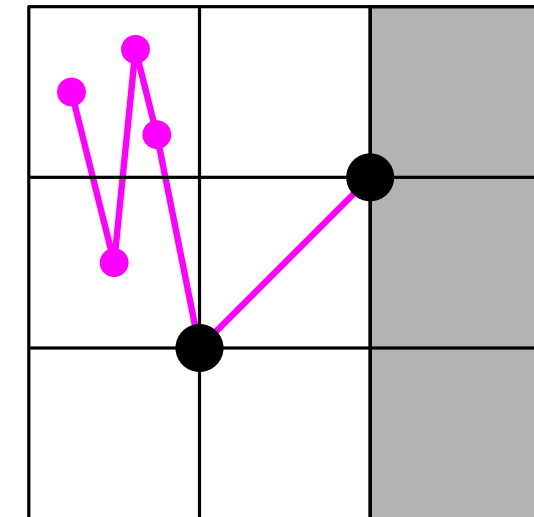
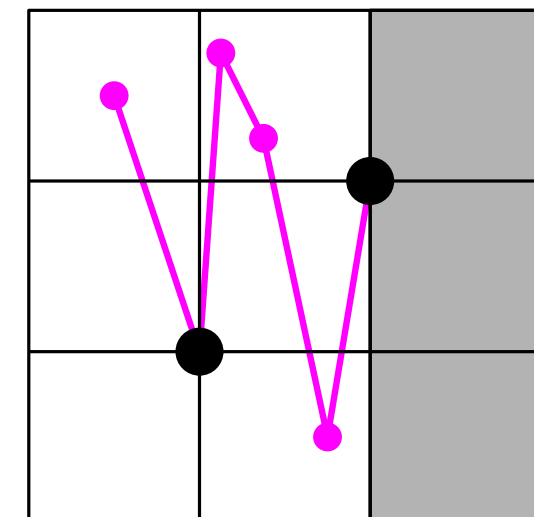
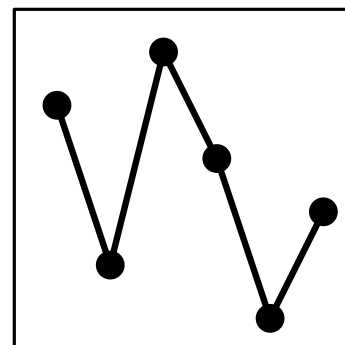
Containers

526413 has a valid preimage.



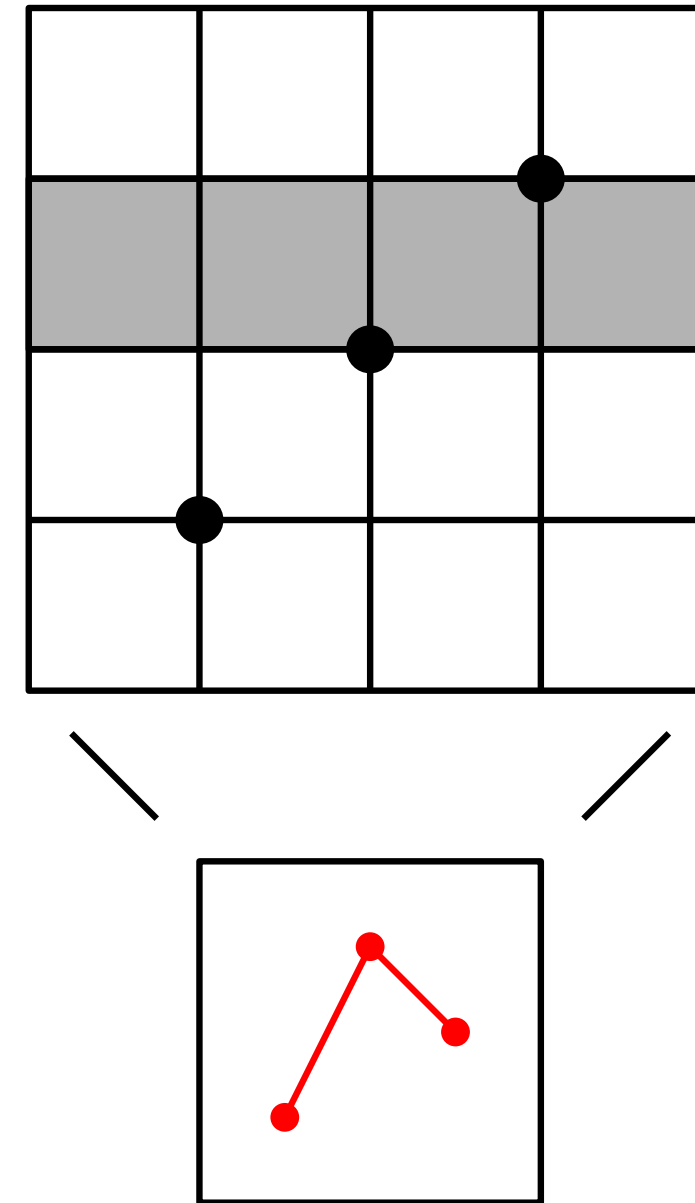
Enumerators

526413 has two valid preimages.



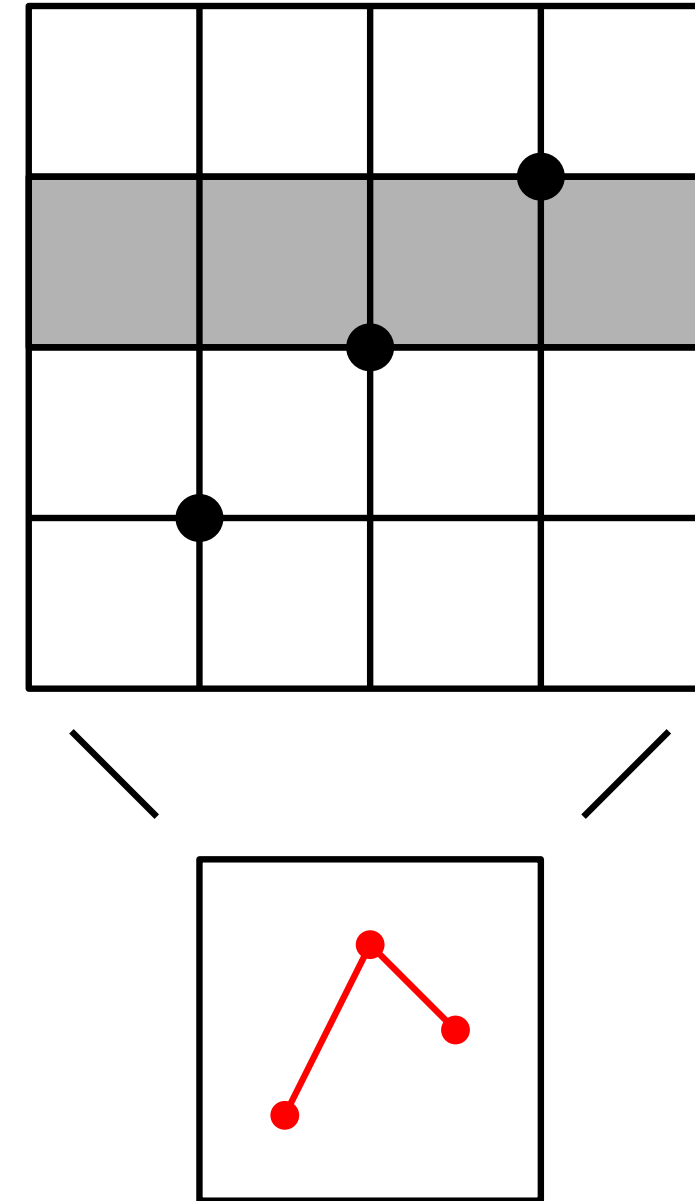
Advantages of Mapped Tilings

- In practice, proof trees using mappings tend to have fewer, more complex, rules than those using normal Telescope.



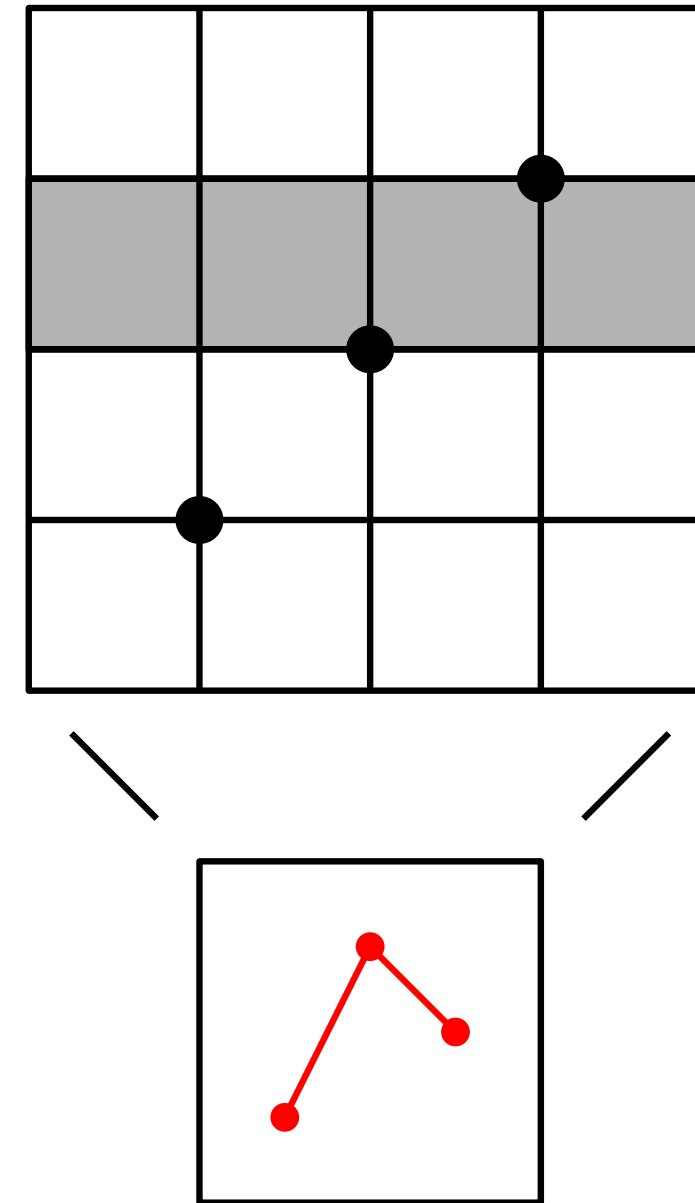
Advantages of Mapped Tilings

- In practice, proof trees using mappings tend to have fewer, more complex, rules than those using normal Telescope.
- Being able to avoid multiple mesh patterns at a time means we could automatically reproduce the results from several papers.



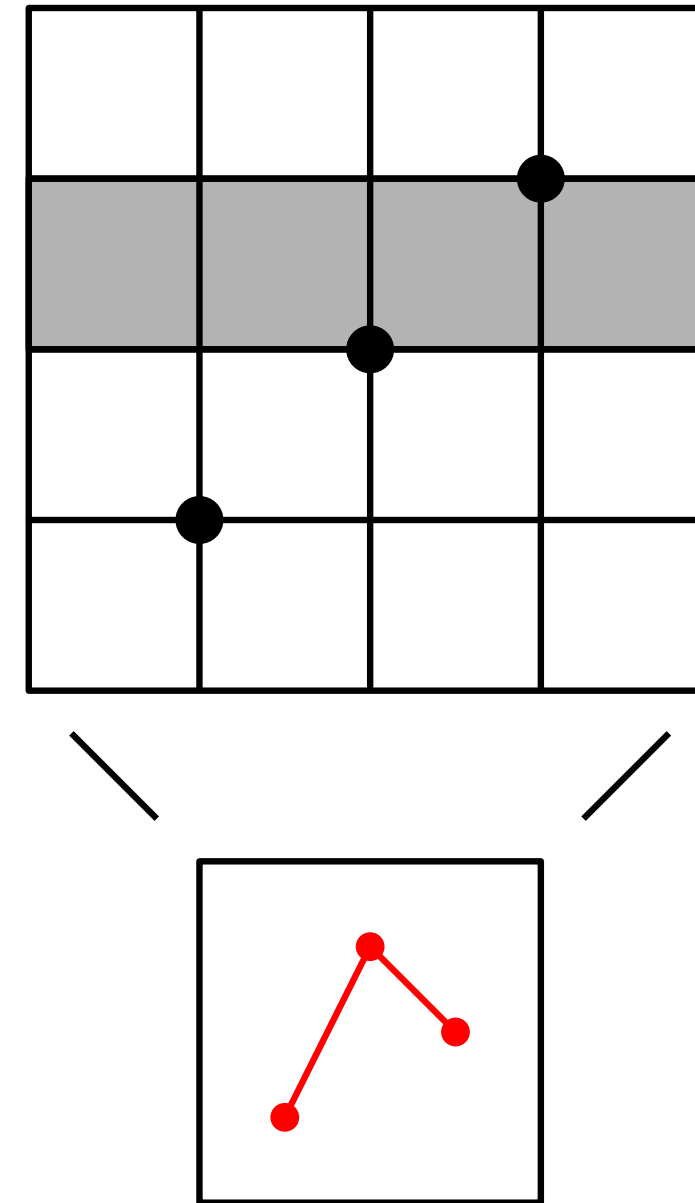
Advantages of Mapped Tilings

- In practice, proof trees using mappings tend to have fewer, more complex, rules than those using normal Telescope.
- Being able to avoid multiple mesh patterns at a time means we could automatically reproduce the results from several papers.
- With a linear combination of mesh pattern enumerating parameters, we can (theoretically) track any permutation statistic. [Brändén, Claeson]



Advantages of Mapped Tilings

- In practice, proof trees using mappings tend to have fewer, more complex, rules than those using normal Telescope.
- Being able to avoid multiple mesh patterns at a time means we could automatically reproduce the results from several papers.
- We have already reproduced proofs and enumerations for more than half of the classes avoiding a pair of Babson-Steingrímsson patterns [Claeson, Mansour].



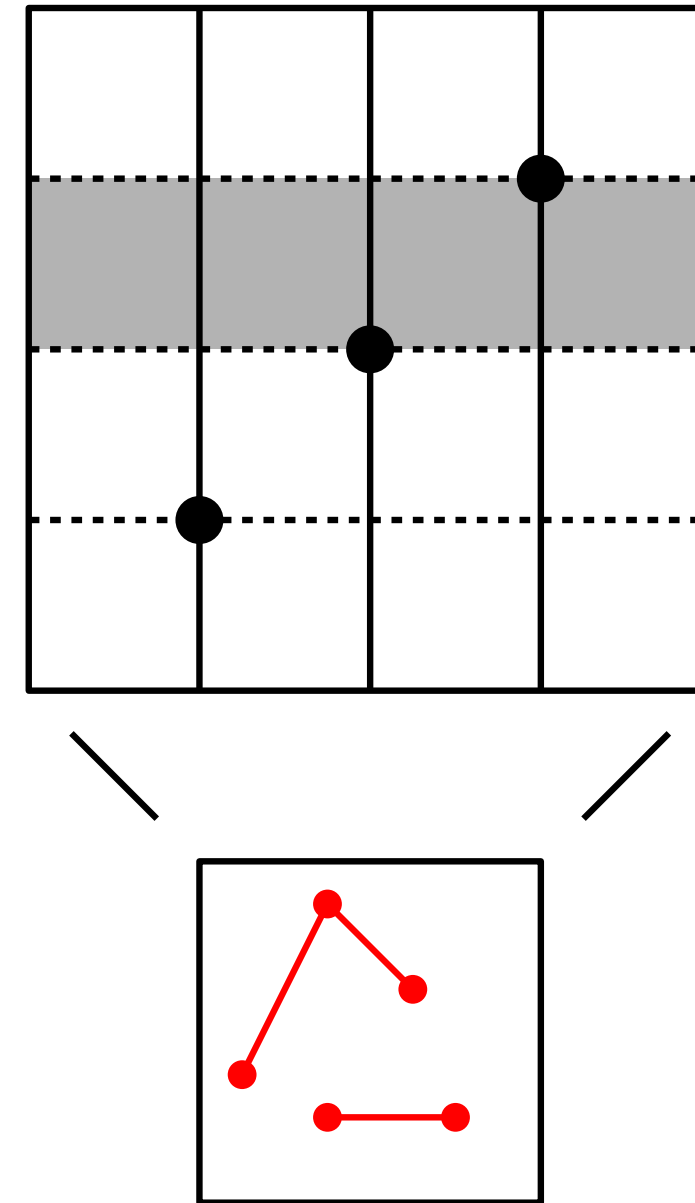


MapplePy Output

Show a tree

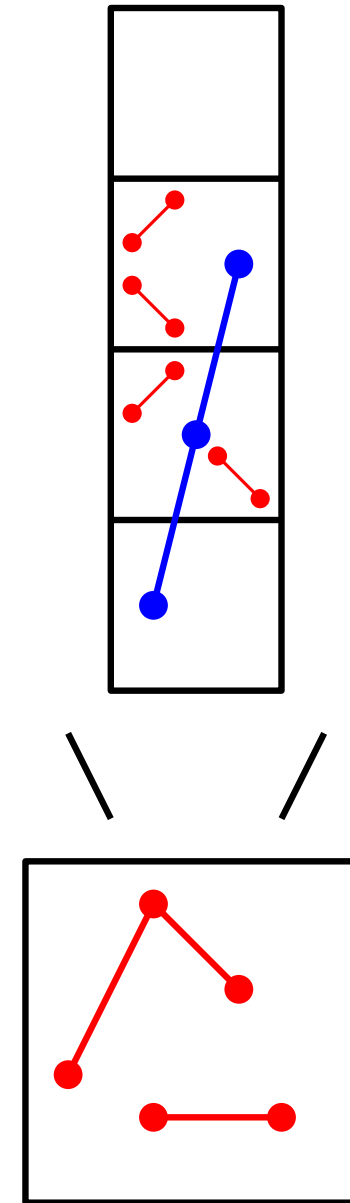
Other Changes From Telescope

- The new framework is built on gridded *Cayley* permutations. The Cayley “Telescope” (without mappings) has been successful in enumerating classical pattern avoidance in Cayley permutations.



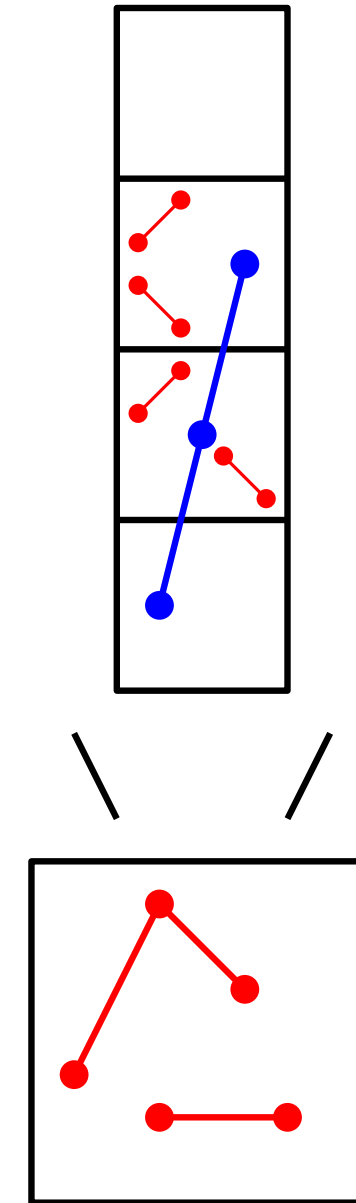
Other Changes From Telescope

- The new framework is built on gridded *Cayley* permutations. The Cayley “Telescope” (without mappings) has been successful in enumerating classical pattern avoidance in Cayley permutations.
- In addition to the existing methods for simplifying a tiling, we created a flexible cleaner structure that will run “simple” inferral strategies in a loop.



Other Changes From Telescope

- The new framework is built on gridded *Cayley* permutations. The Cayley “Telescope” (without mappings) has been successful in enumerating classical pattern avoidance in Cayley permutations.
- In addition to the existing methods for simplifying a tiling, we created a flexible cleaner structure that will run “simple” inferral strategies in a loop.
- There are certain strategies that can only be applied to mappings.





References

Albert, Michael H., Christian Bean, Anders Claesson, Emile Nadeau, Jay Pantone, and Henning Ulfarsson. 2026. “COMBINATORIAL EXPLORATION: AN ALGORITHMIC FRAMEWORK FOR ENUMERATION.” *Memoirs of the American Mathematical Society* 317.

Claesson, Anders, and Toufik Mansour. 2005. “Enumerating Permutations Avoiding a Pair of Babson-Steingrímsson Patterns.” *Ars Combinatoria* 77 (October). <http://www.combinatorialmath.ca/arscombinatoria/vol77.html>.