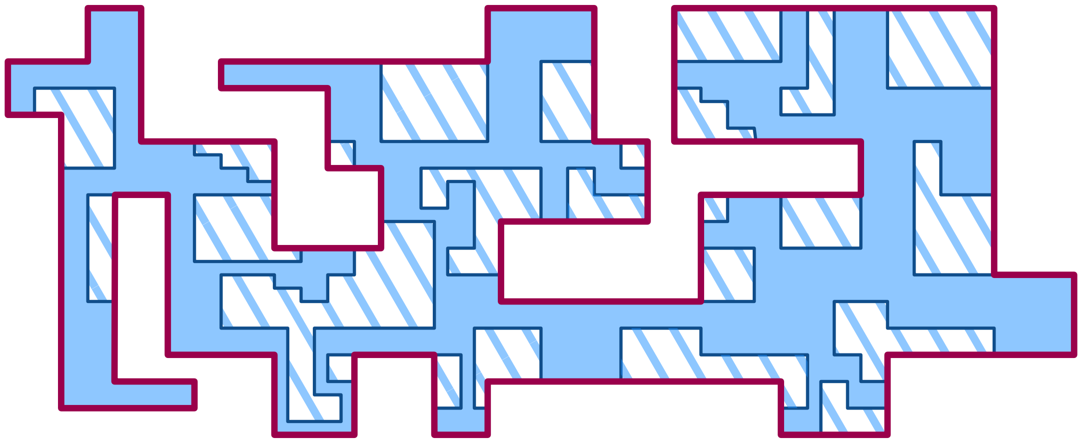
Tight Rectilinear Hulls of Simple Polygons

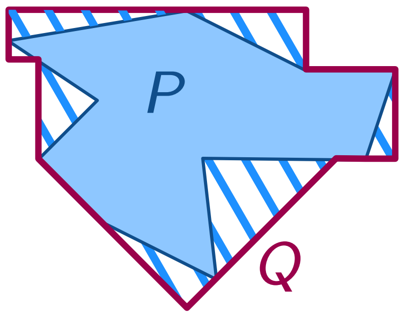


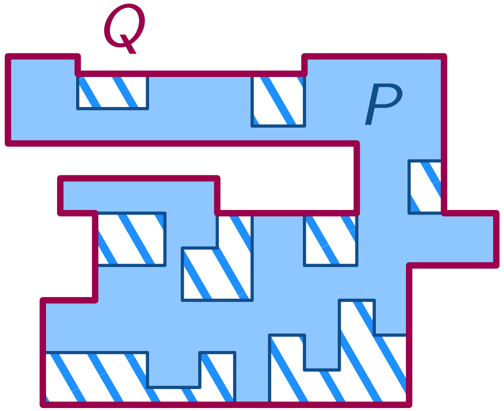
Annika Bonerath, Jan-Henrik Haunert and Benjamin Niedermann

Institute of Geodesy und Geoinformation,
University of Bonn

Motivation

Main Goal:
Simplification of a polygon P with a polygon Q.





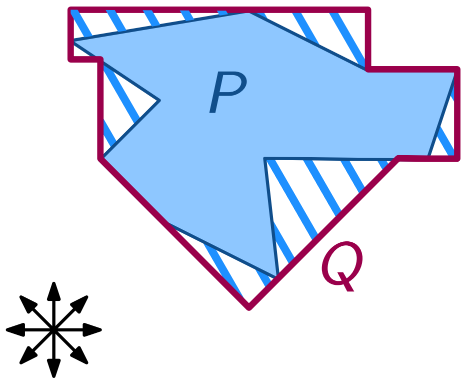
Motivation

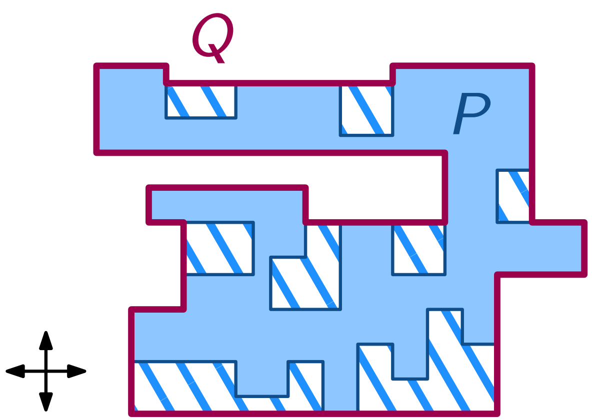
Main Goal:
Simplification of a polygon P with a polygon Q.

Requirements for Q:

- simple
- $\mathcal{C}$-oriented
- contains P
- cannot be shrunk

(formalization follows on the next slides)





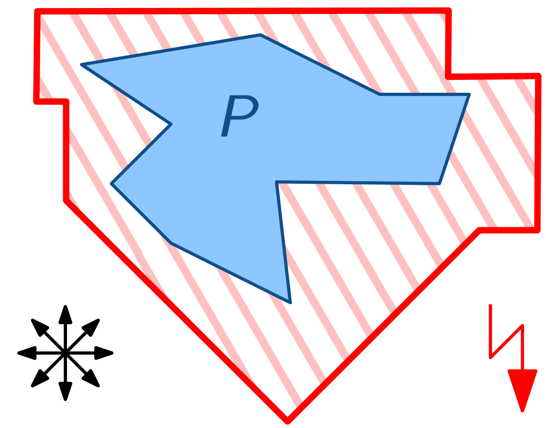
Motivation

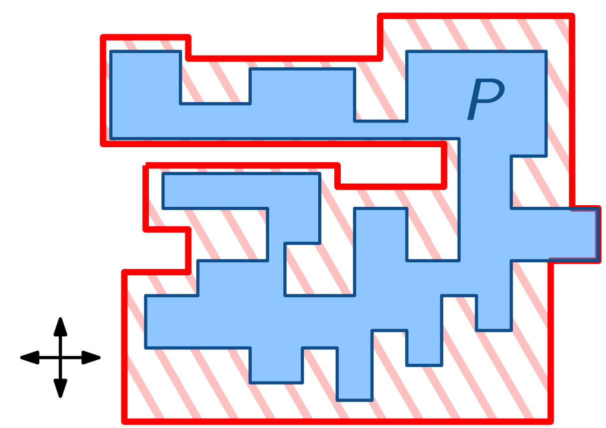
Main Goal:
Simplification of a polygon P with a polygon Q.

Requirements for Q:

- simple
- $\mathcal{C}$-oriented
- contains P
- cannot be shrunk

(formalization follows on the next slides)





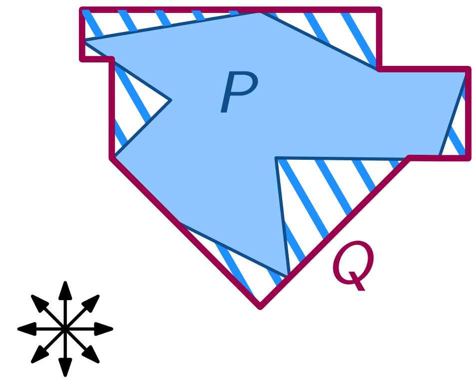
Motivation

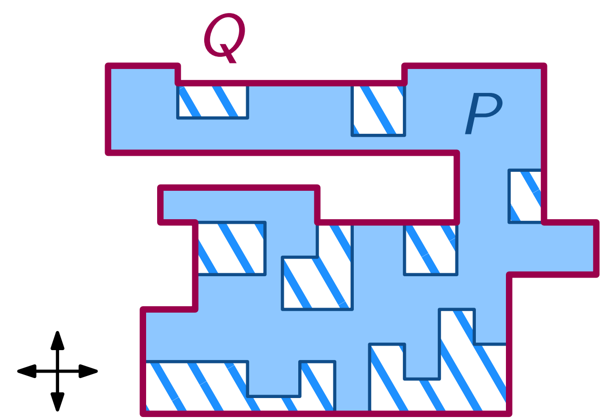
Main Goal:
Simplification of a polygon P with a polygon Q.

Requirements for Q:
- simple
- $\mathcal{C}$-oriented
- contains P
- cannot be shrunk

(formalization follows on the next slides)

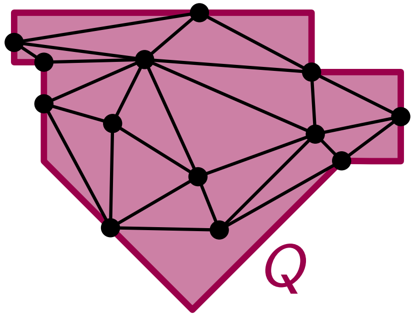
Optimization Goal:
few bends, small area, short length





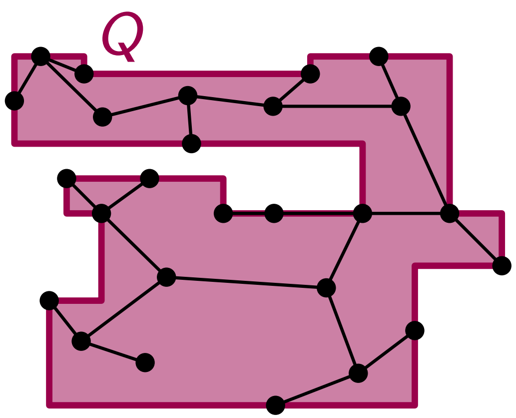
Motivation

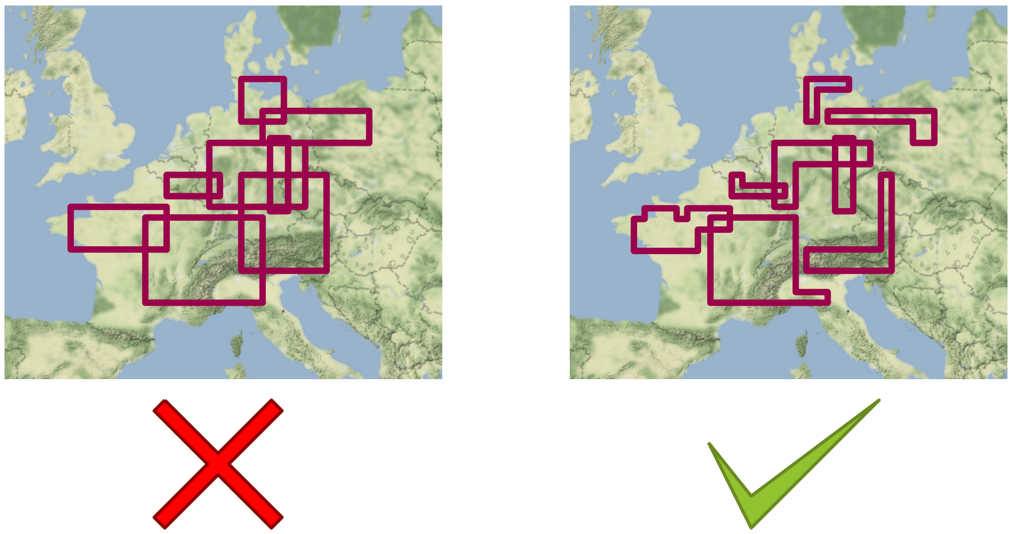
Main Goal:
Simplification of a polygon P with a polygon Q.



Application:

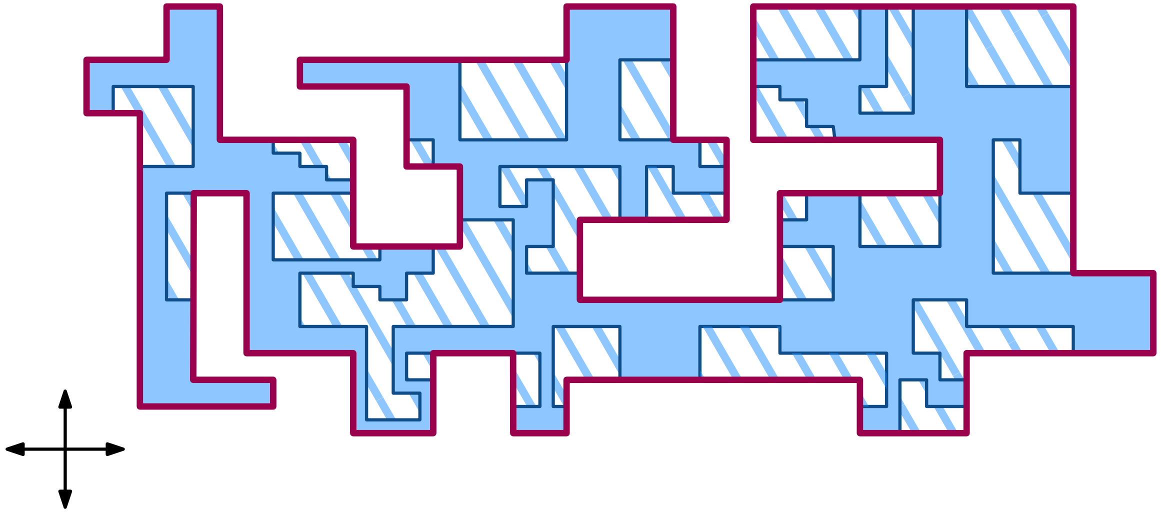
- schematization of plane graph drawings
- travel-time maps that visualize reachable parts in a road network
- schematic representation of point sets, instead of using bounding boxes as usually done in data management systems





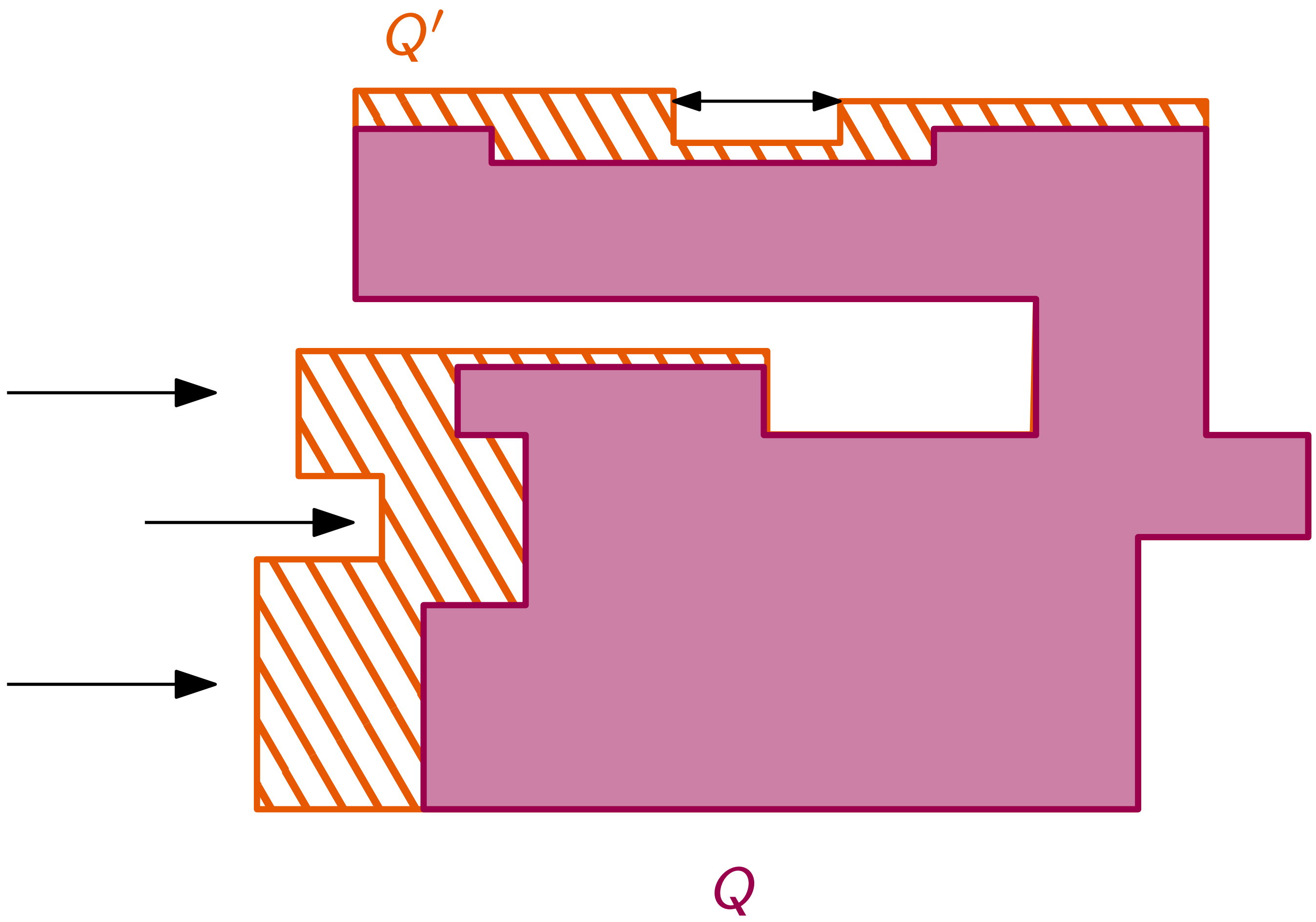
This Paper

Restriction to rectilinear simple input and output polygons.



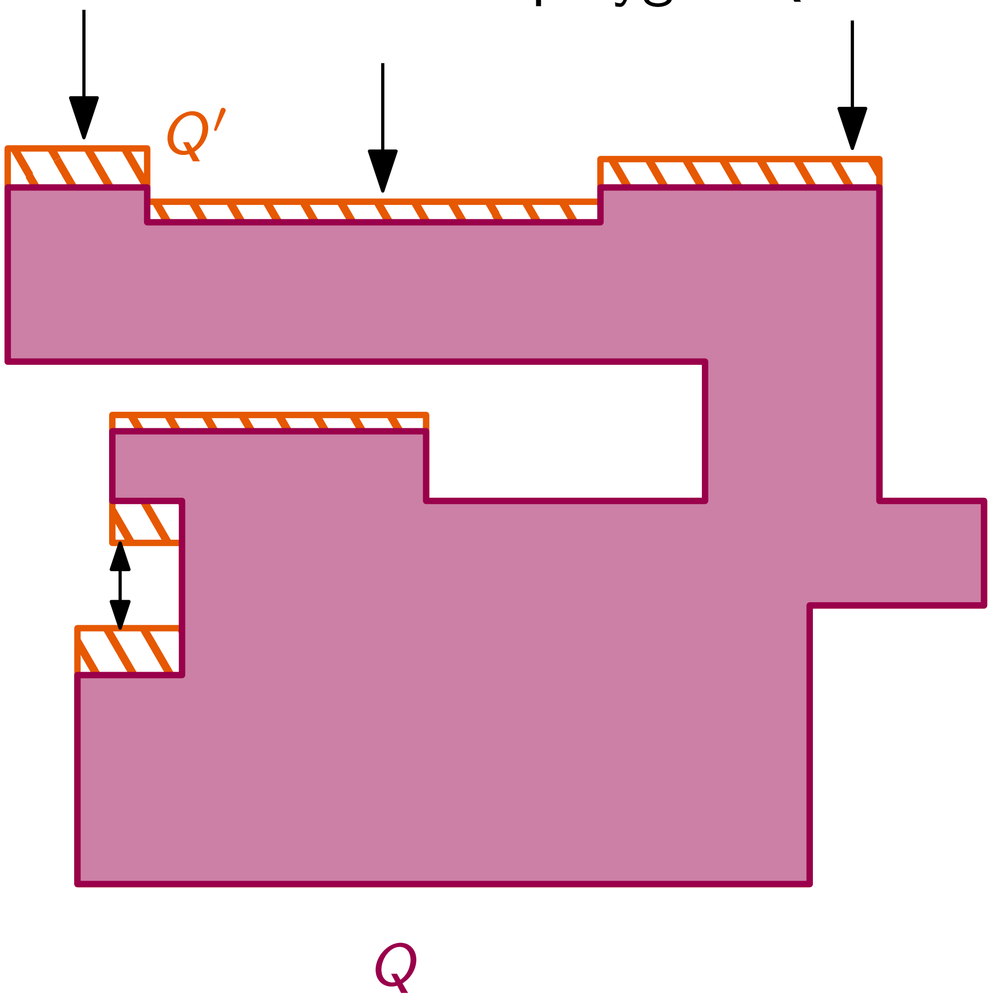
Formalization of Tight Hulls

Definition:
The polygon Q' is a *linear distortion* of Q if each edge of Q' can be scaled and translated such that the polygon Q results.



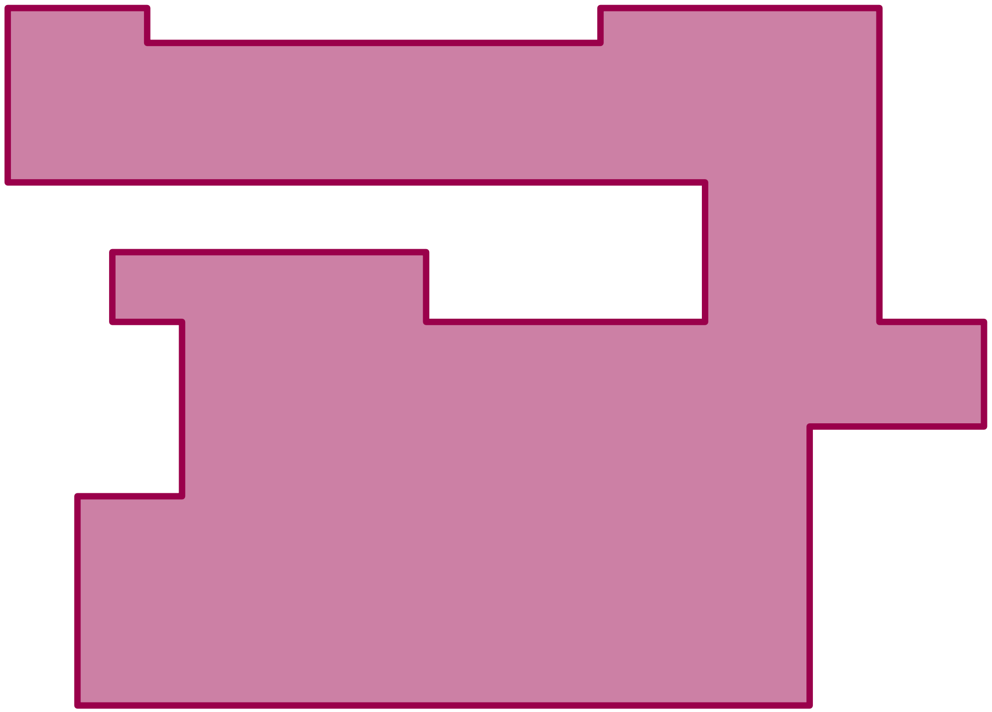
Formalization of Tight Hulls

Definition:
The polygon Q' is a *linear distortion* of Q if each edge of Q' can be scaled and translated such that the polygon Q results.



Formalization of Tight Hulls

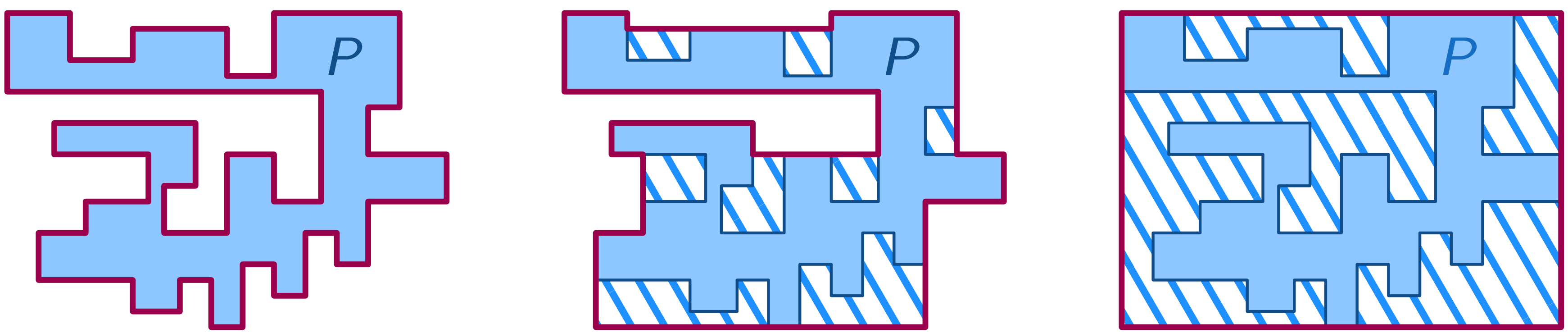
Definition:
The polygon Q' is a *linear distortion* of Q if each edge of Q' can be scaled and translated such that the polygon Q results.



$Q' = Q$

Formalization of Tight Hulls

Definition:
A **simple** polygon Q is a *tight hull* of another polygon P if Q contains P and there is no linear distortion of Q that lies in Q and contains P.



$Q = P$ $\quad$ Q $\quad$ $Q =$ bounding box of P

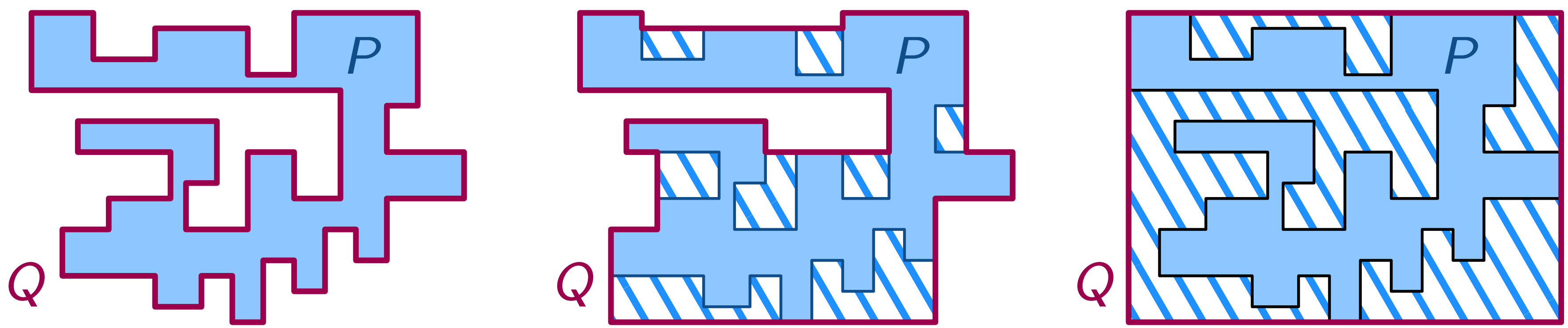
What is a good tight hull?

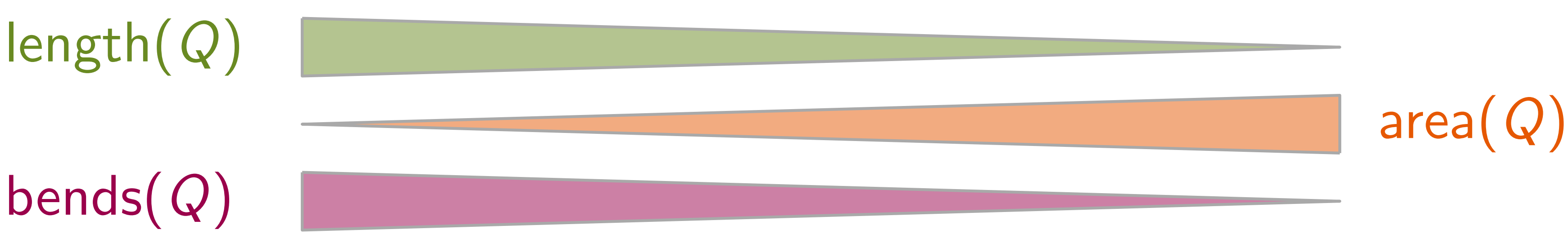
Formalization of Tight Hulls

Definition:
The tight hull Q of P is α-*optimal* if Q minimizes

$$\text{cost}(Q) = \alpha_1 \cdot \text{length}(Q) + \alpha_2 \cdot \text{area}(Q) + \alpha_3 \cdot \text{bends}(Q)$$

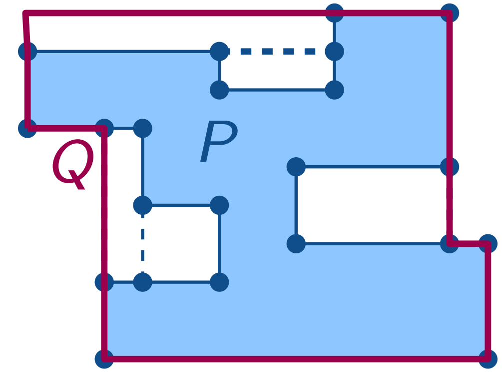
over all tight hulls Q'.





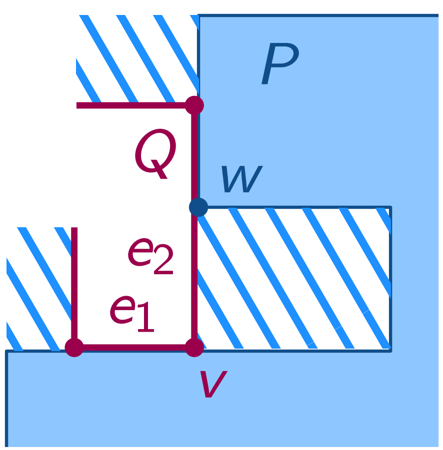
Structural Properties

Lemma 1:
Every vertex of Q on P is a vertex of the maximally subdivided P.

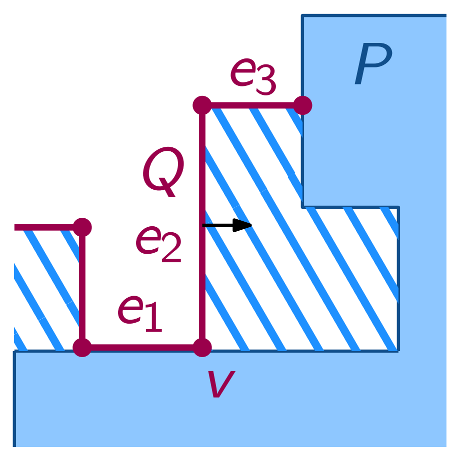


Idea of Proof:

v is a vertex of P



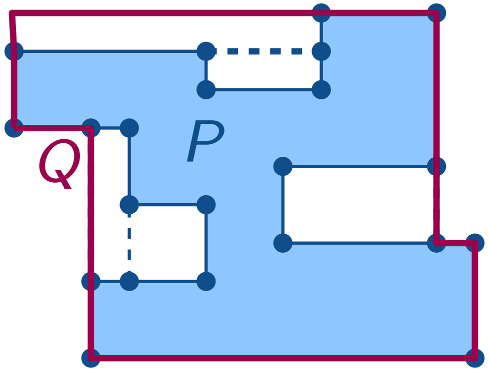
v is not a vertex of P



but then Q is not tight (scale e_1 and e_3)

Structural Properties

Lemma 1:
Every vertex of Q on P is a vertex of the maximally subdivided P.



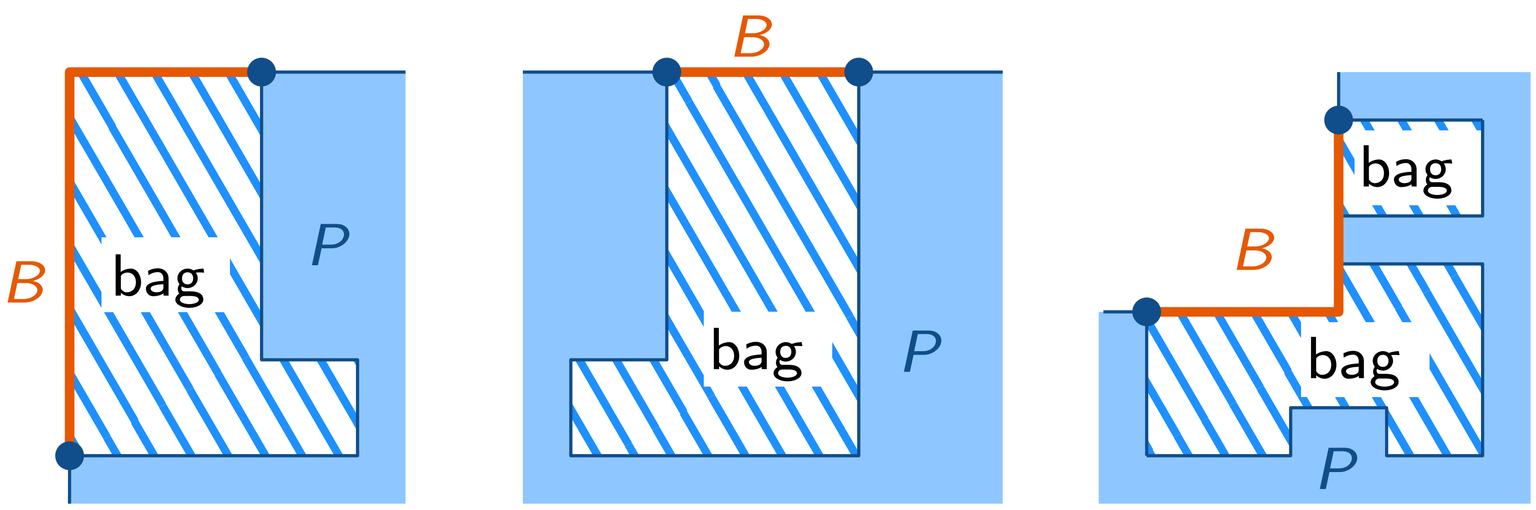
$\Rightarrow$ use vertices of P for the computation of Q

Structural Properties

Definition:
The polyline B is a *bridge* if,

- it consists of one or two incident line segments forming an "L"
- it starts and ends at vertices of P.

Definition:
The region enclosed by B and the polyline of P connecting the same vertices as B is the *bag* of B.



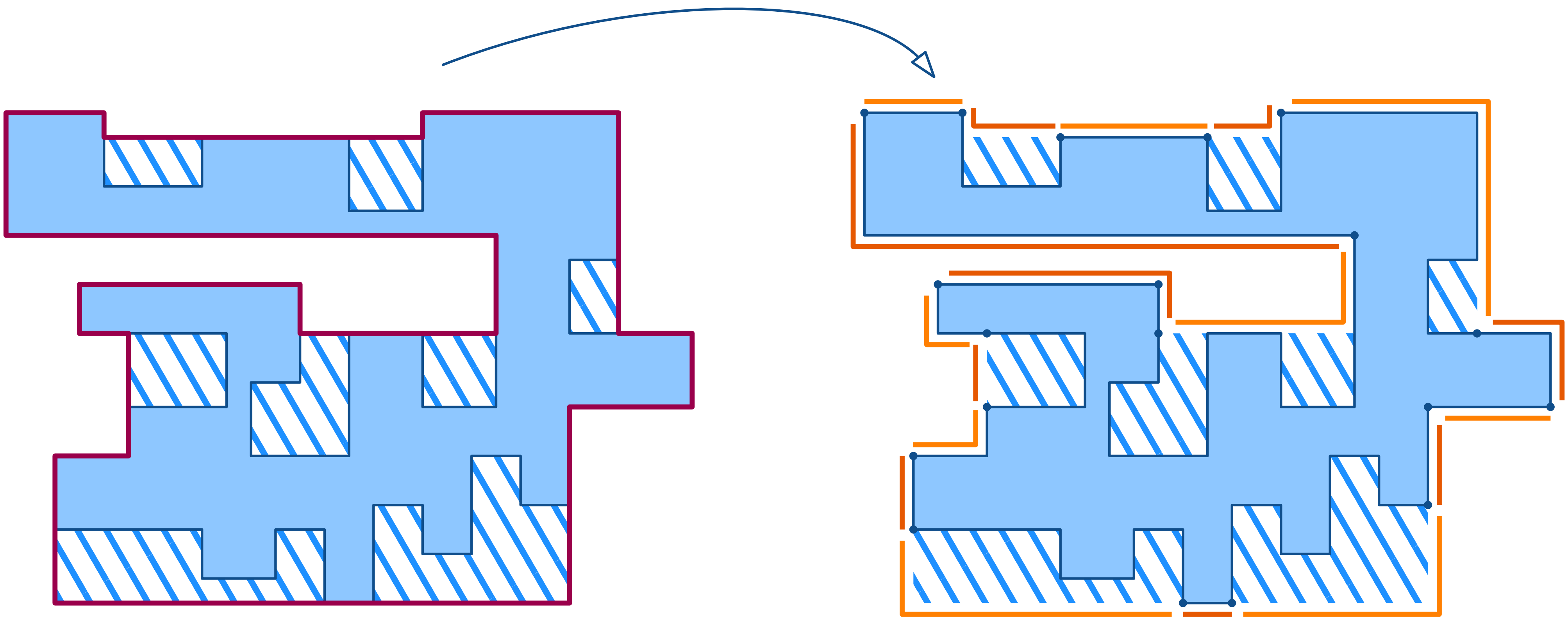
Structural Properties

Definition:

The polyline B is a *bridge* if,

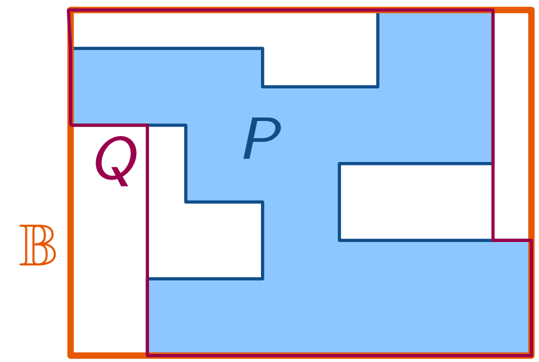
- it consists of one or two incident line segments forming an "L"
- it starts and ends at vertices of P.

$\Rightarrow$ every tight hull can be represented by a set of bridges



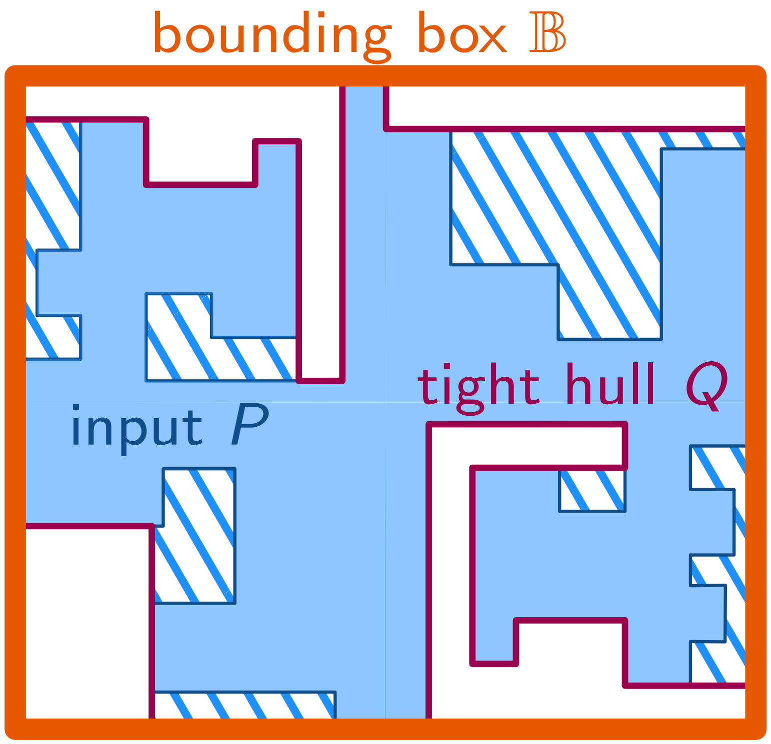
Structural Properties

Lemma 2:
The bounding box $\mathbb{B}$ of P is a tight hull and any other tight hull of P is contained in $\mathbb{B}$.



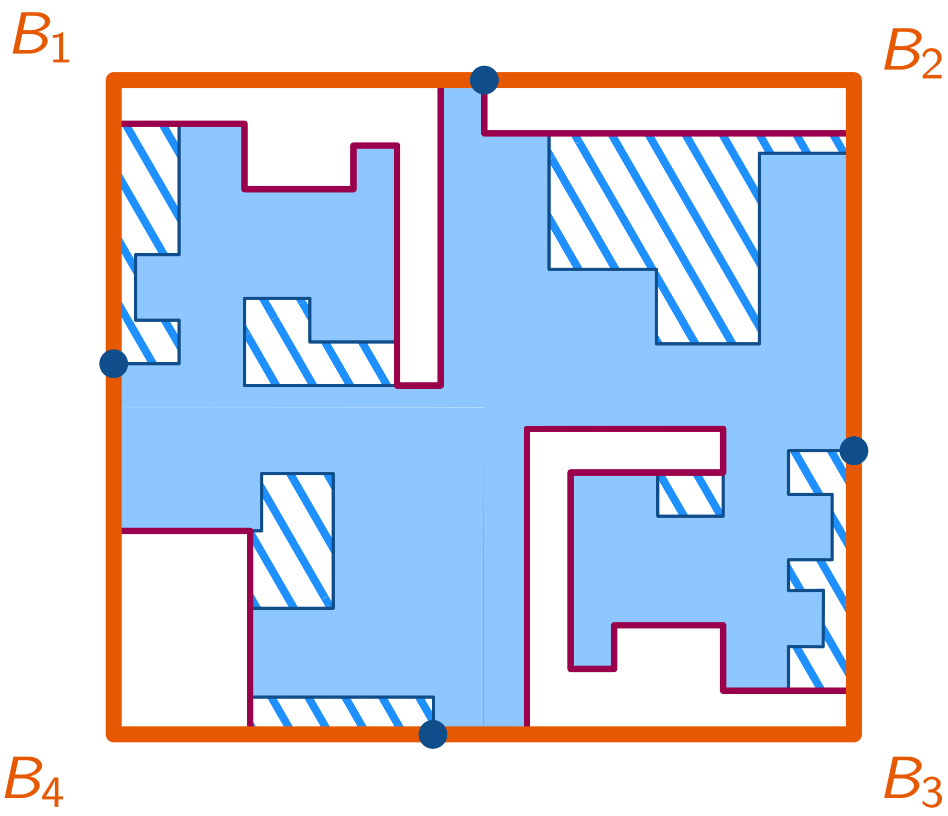
Idea:
carve into the bounding box $\mathbb{B}$ to generate any tight hull

Decomposition



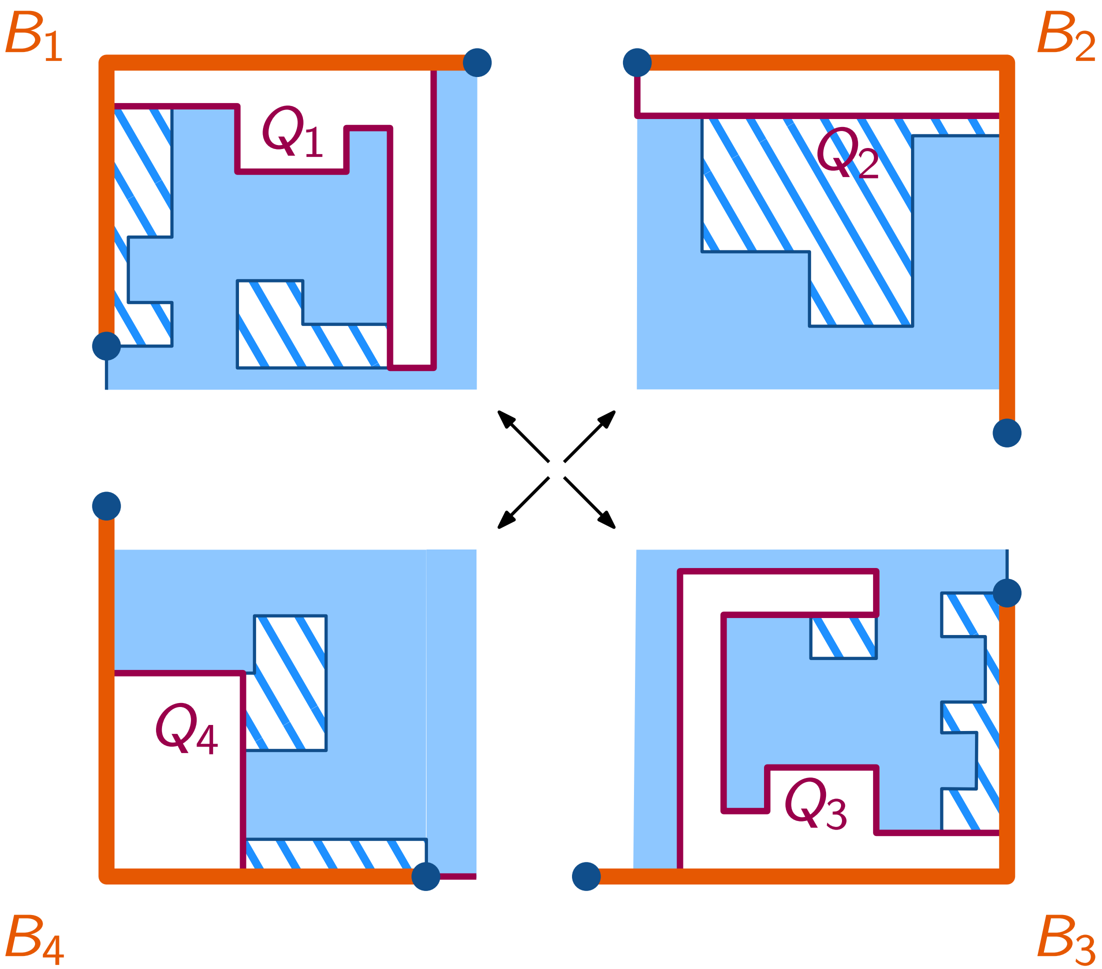
Decomposition

Decompose $\mathbb{B}$ into four independent subinstances defined by bridges B_1, B_2, B_3 and B_4.



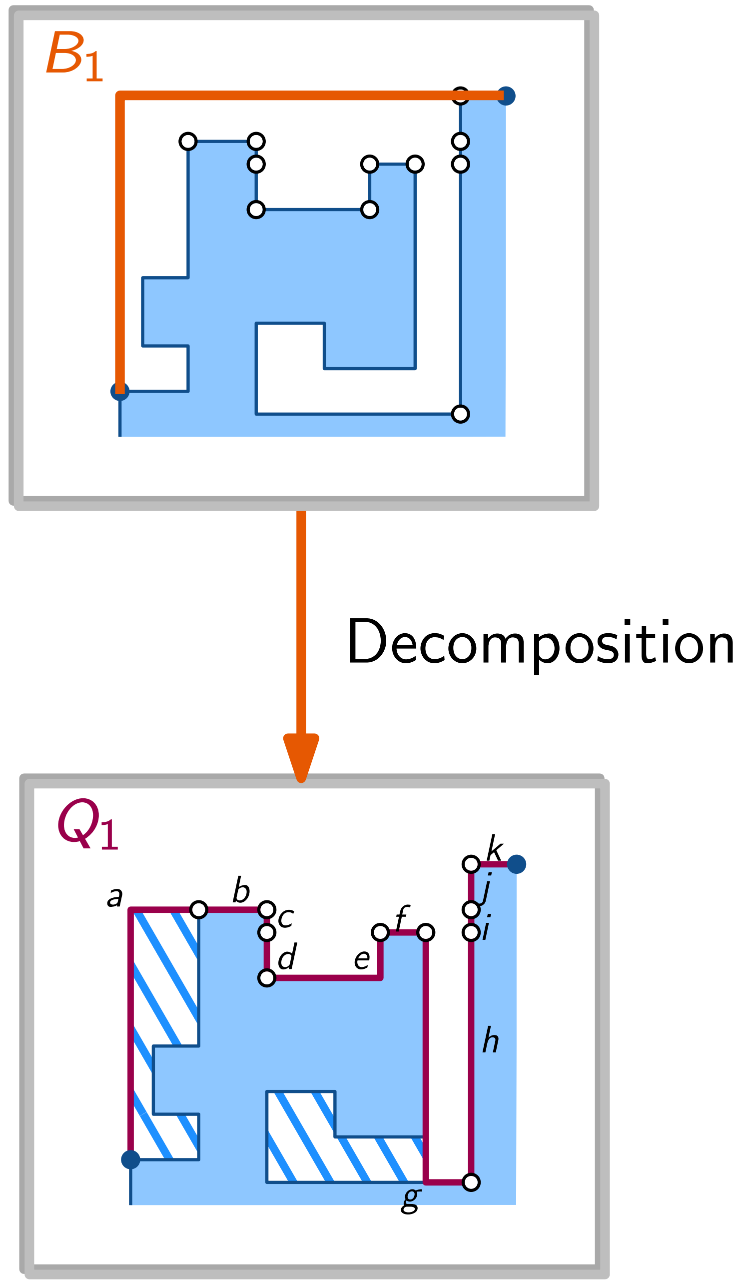
Decomposition

Decompose $\mathbb{B}$ into four independent subinstances defined by bridges B_1, B_2, B_3 and B_4.



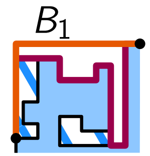
Idea:
Solve instances independently and compose solutions.

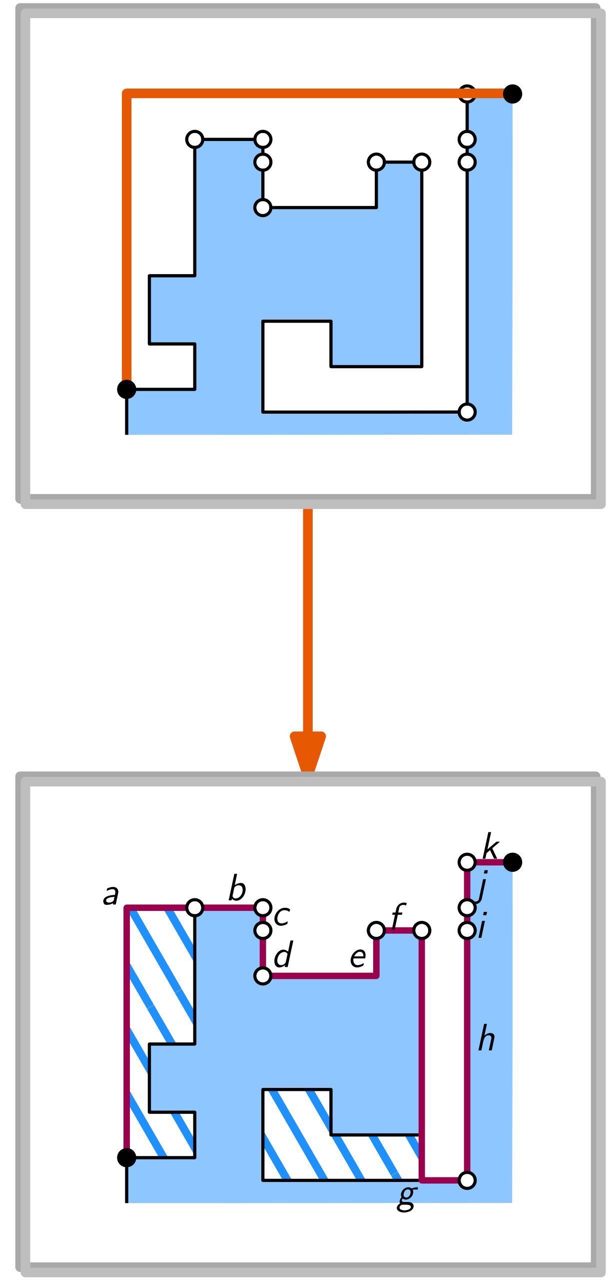
Decomposition



Decomposition

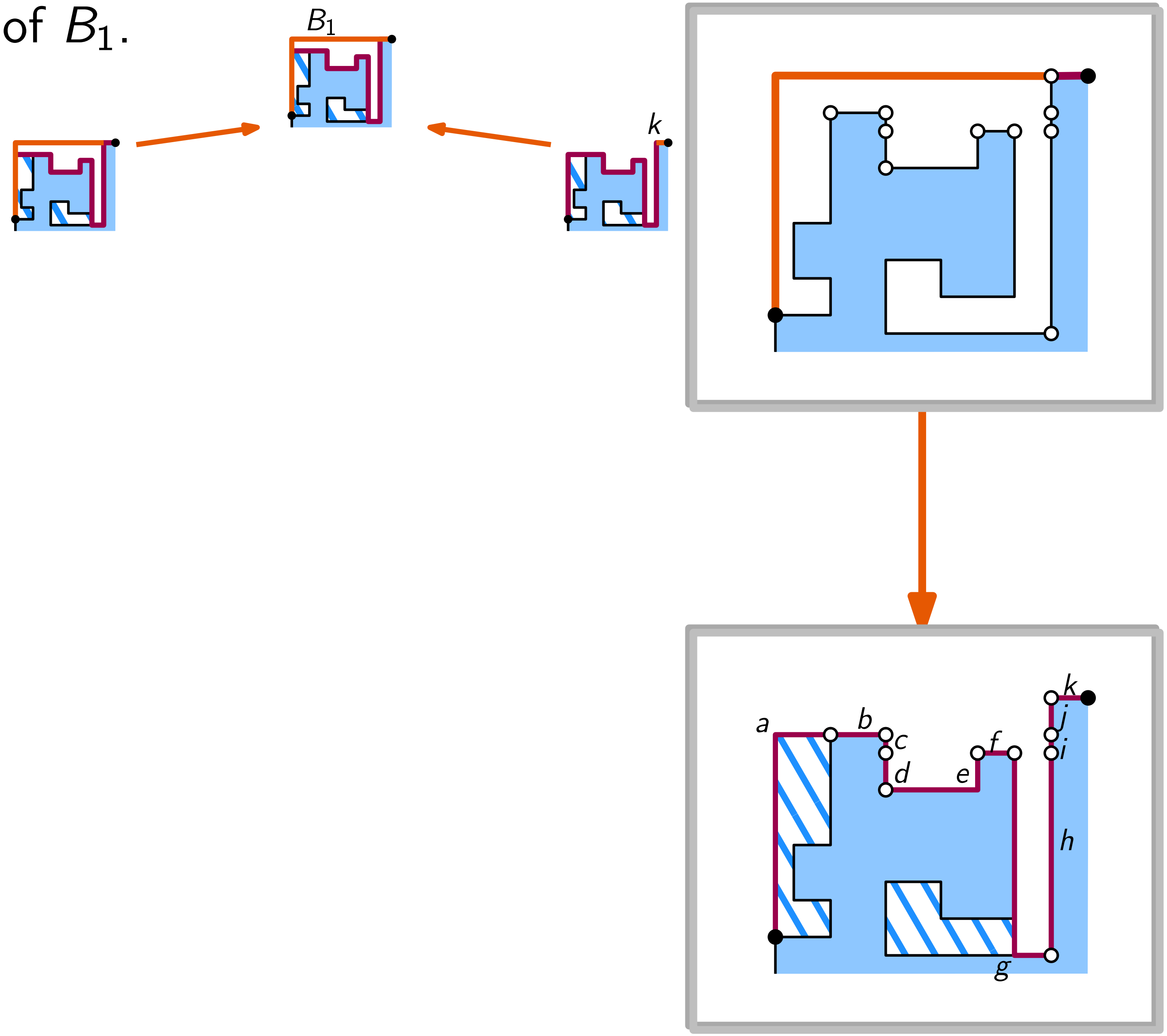
Decomposition tree of B_1.





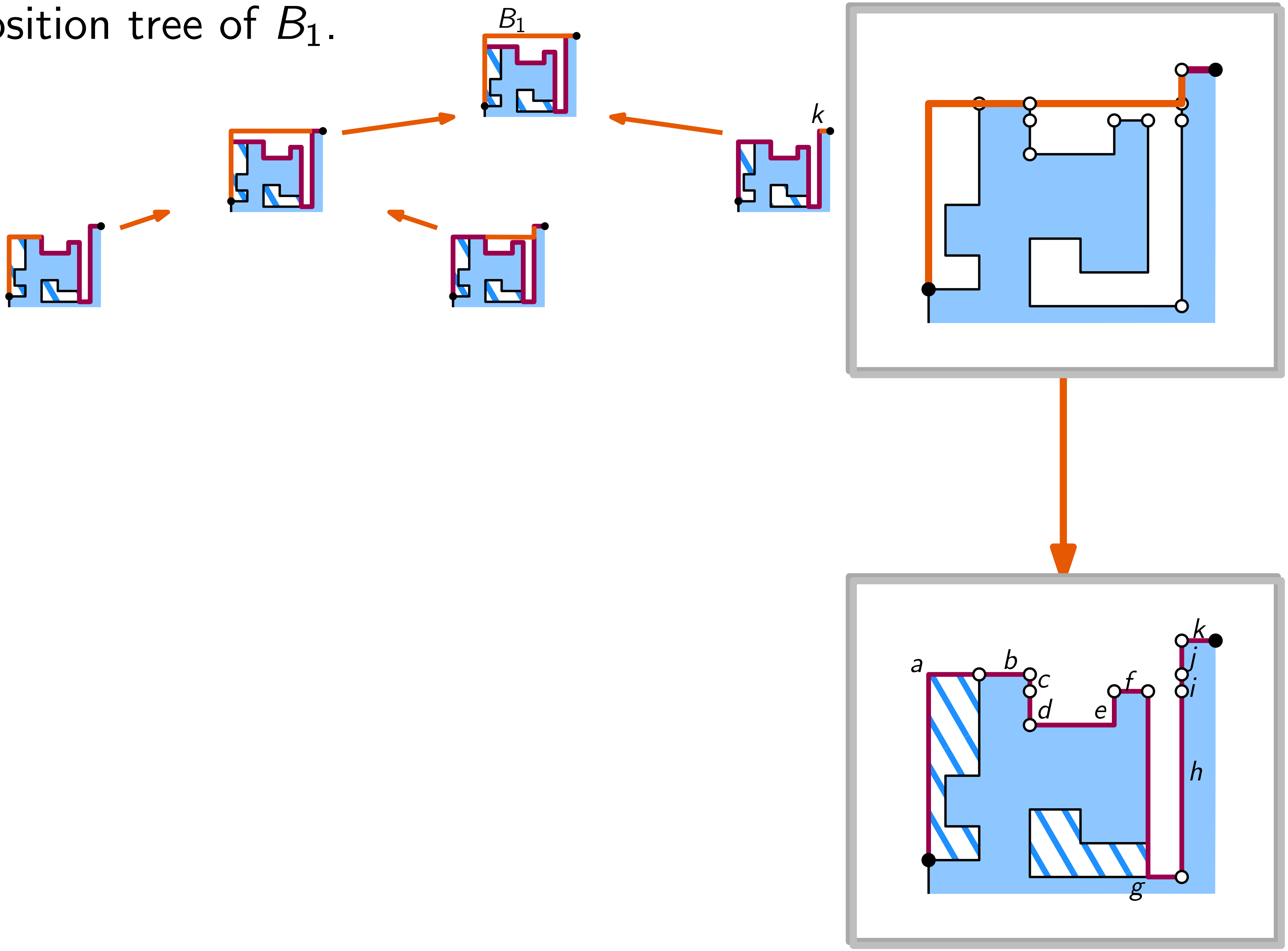
Decomposition

Decomposition tree of B_1.



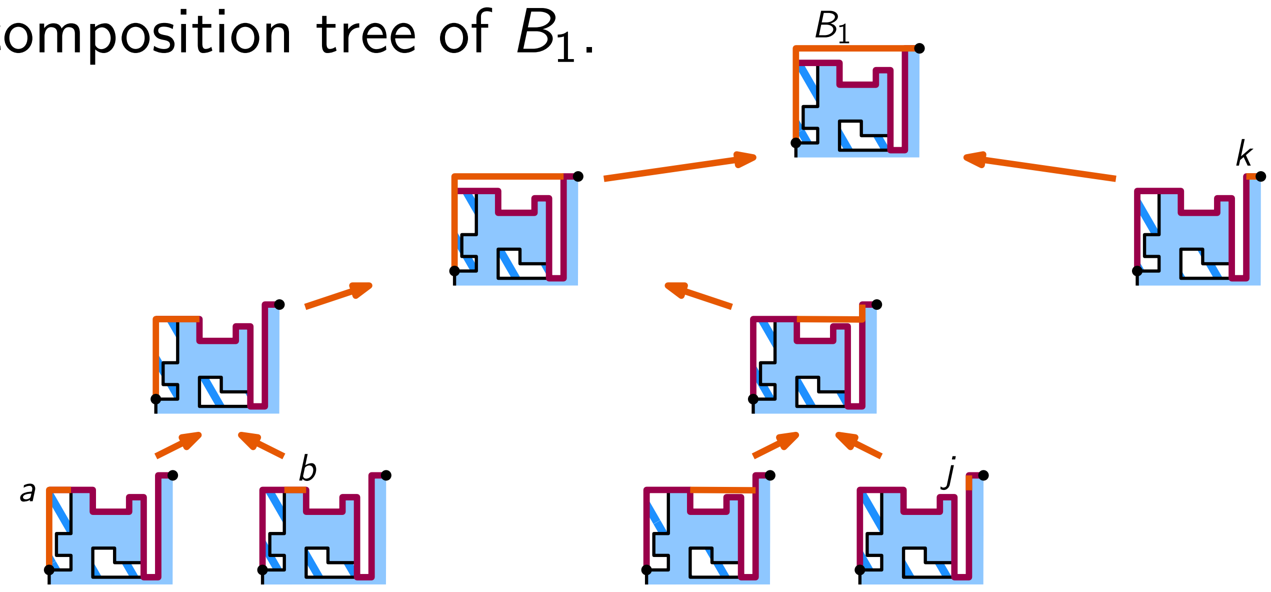
Decomposition

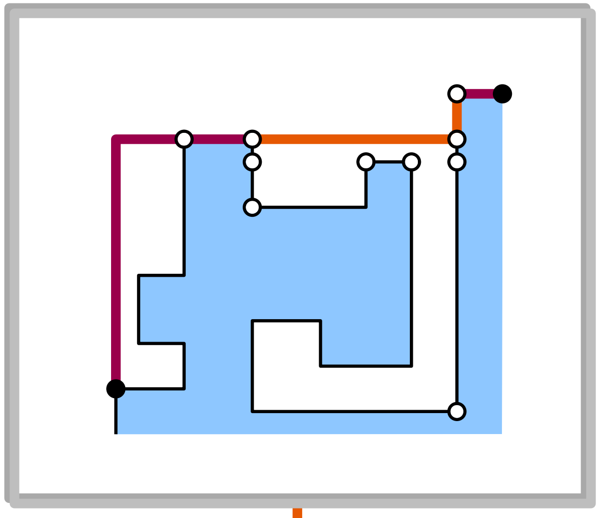
Decomposition tree of B_1.

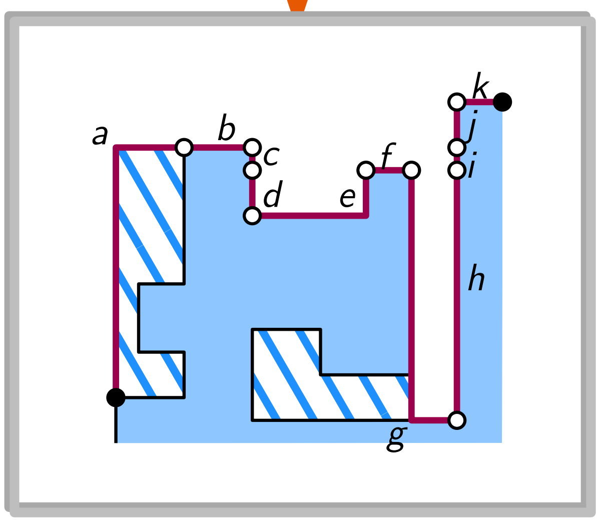


Decomposition

Decomposition tree of B_1.

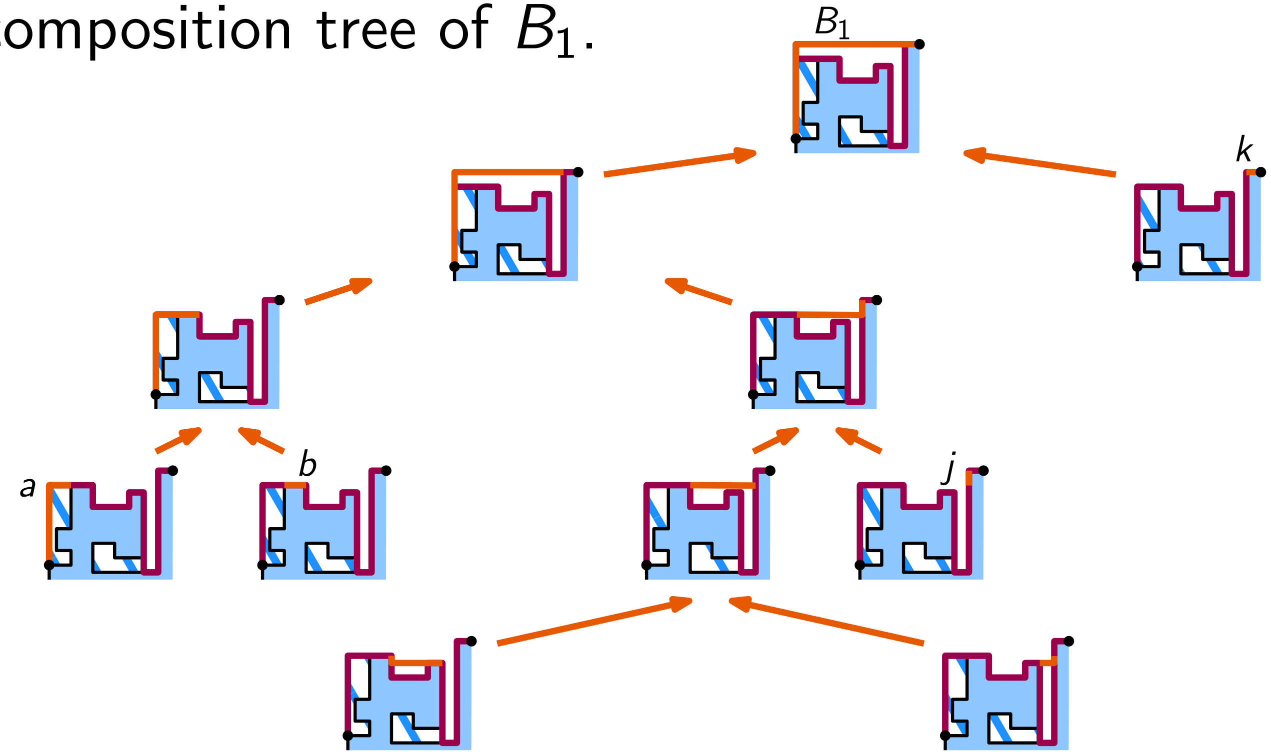


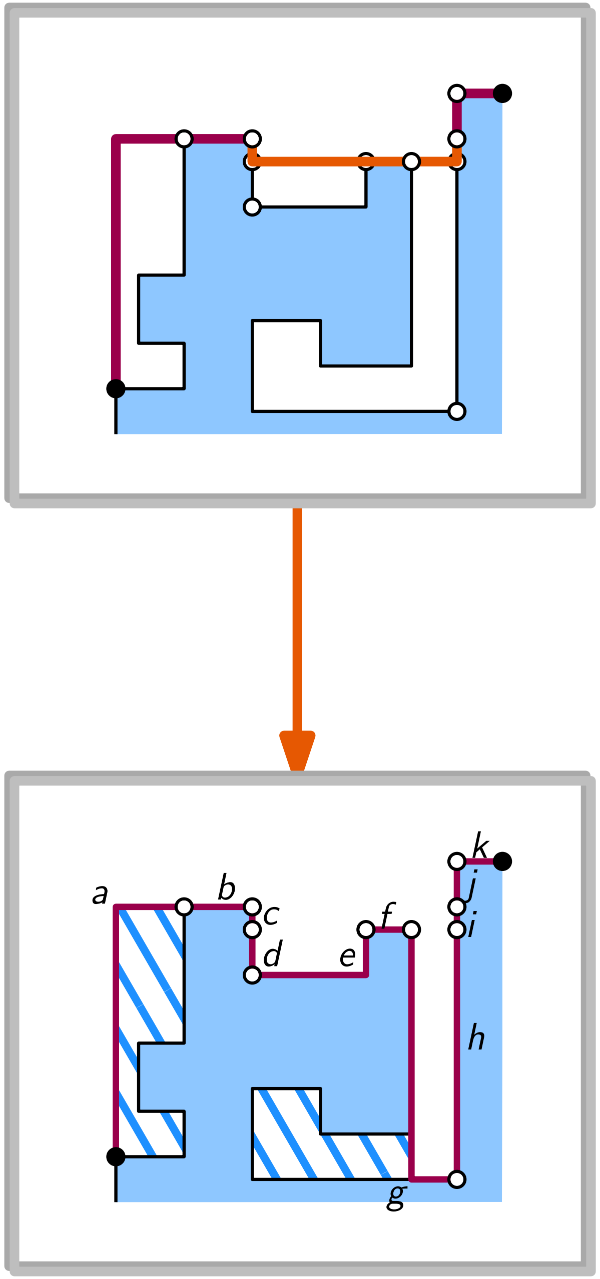




Decomposition

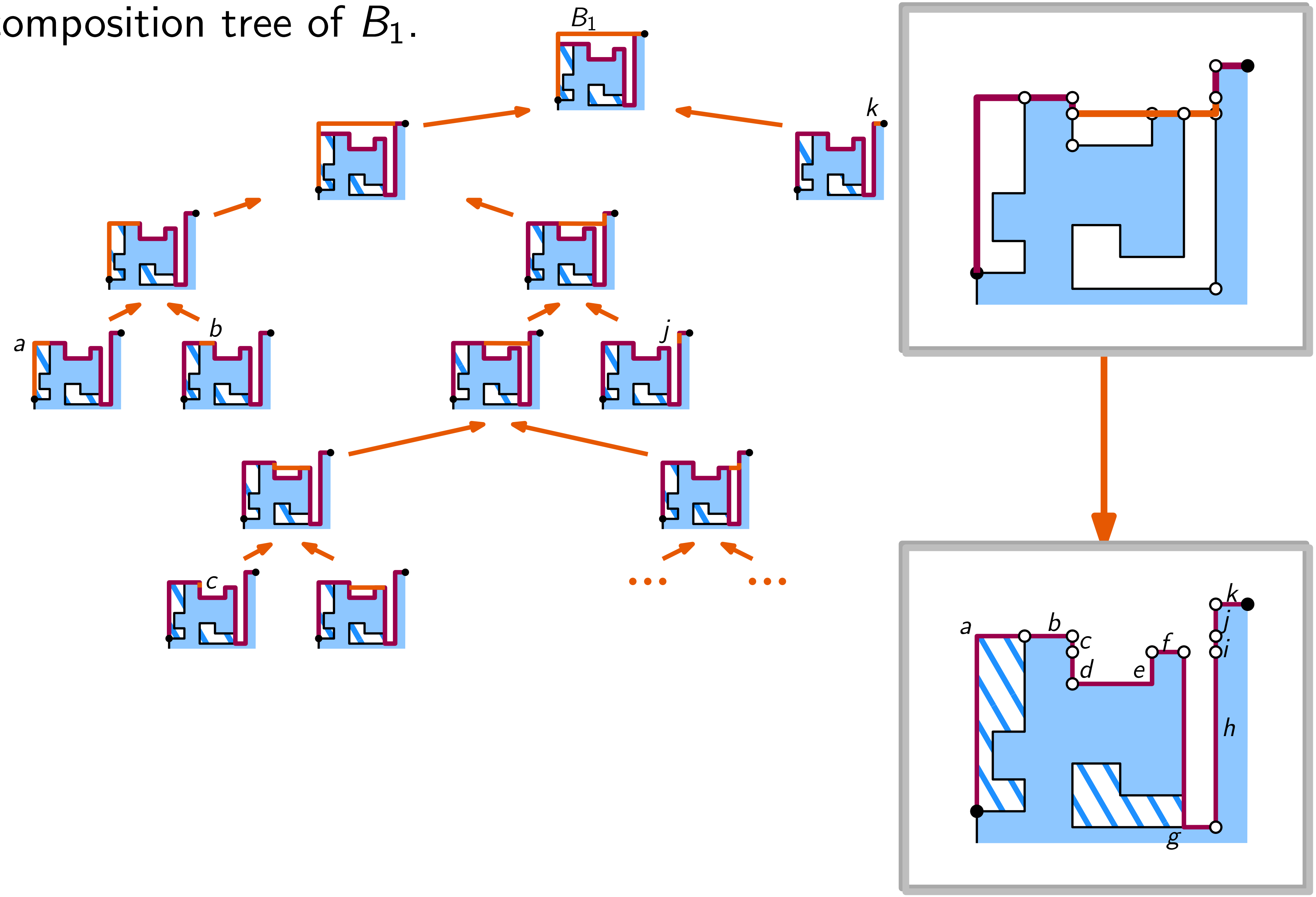
Decomposition tree of B_1.





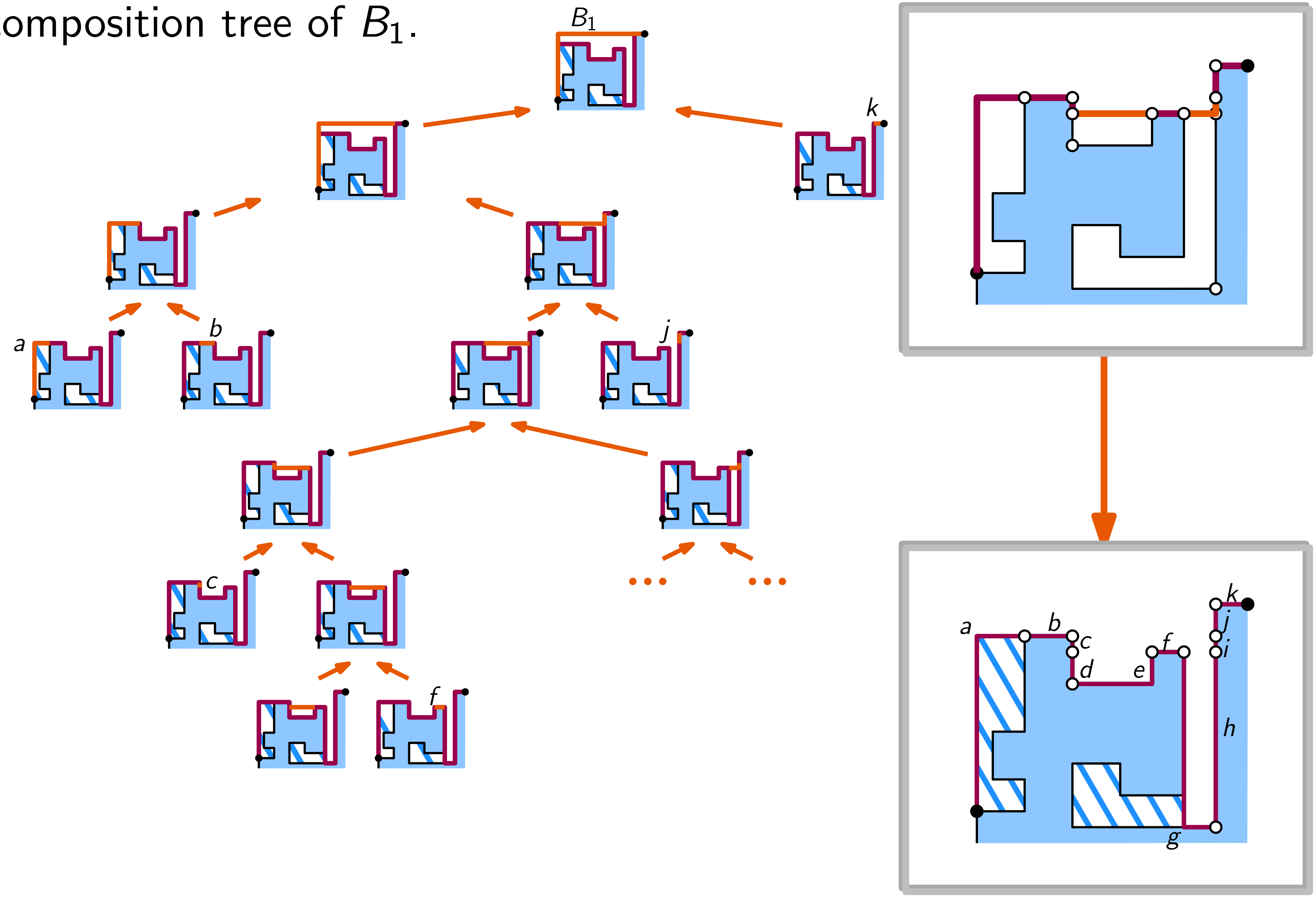
Decomposition

Decomposition tree of B_1.



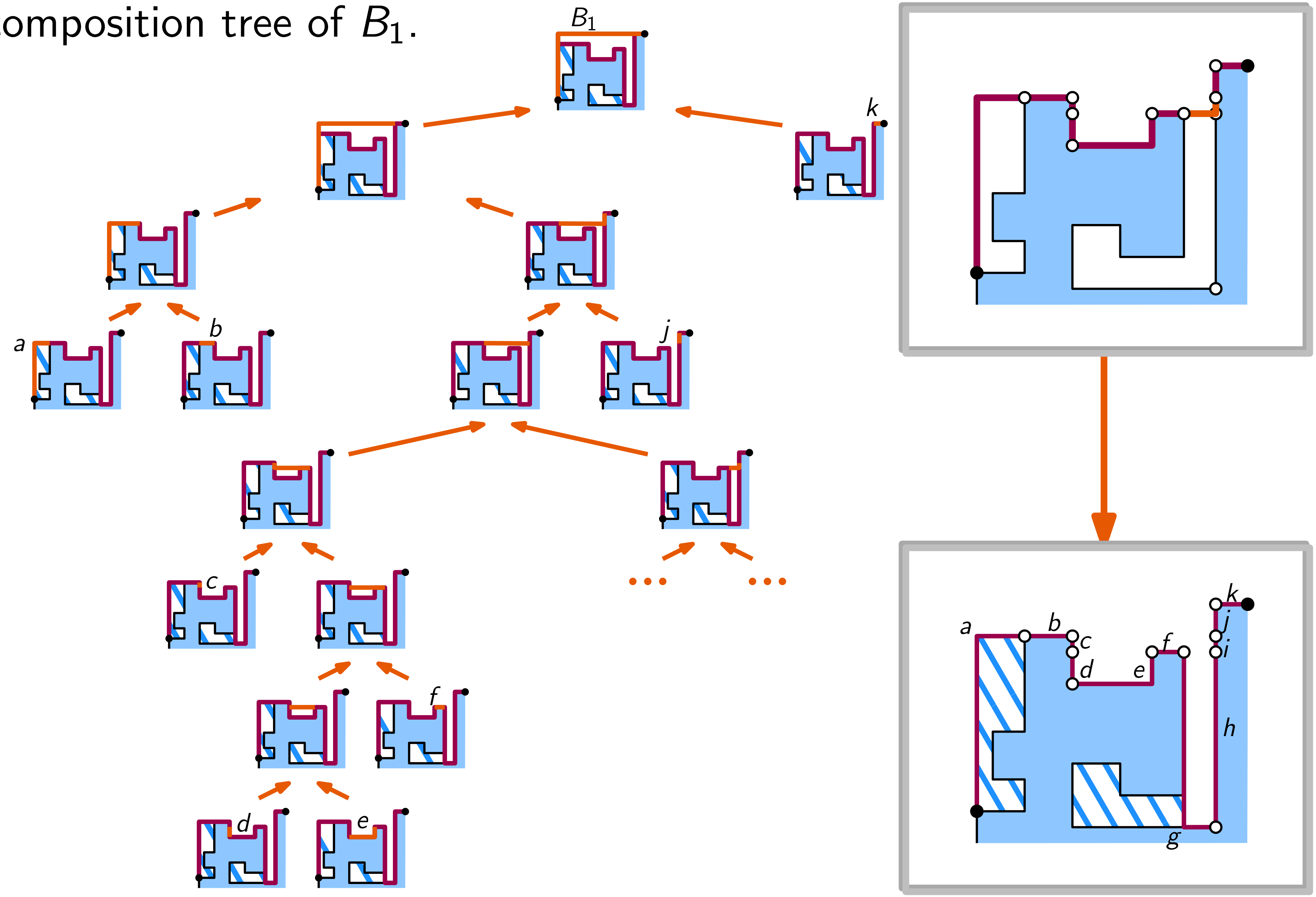
Decomposition

Decomposition tree of B_1.



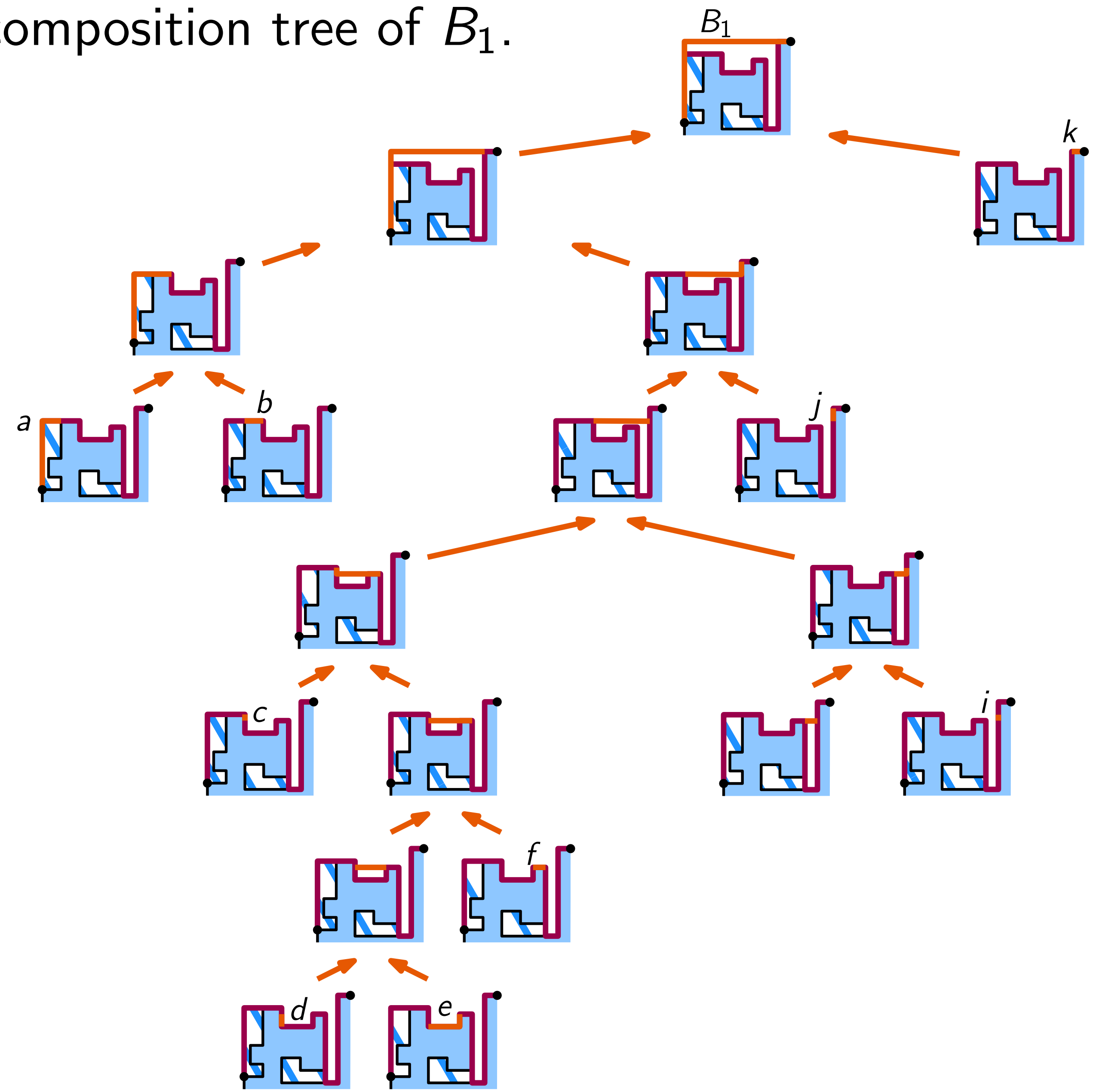
Decomposition

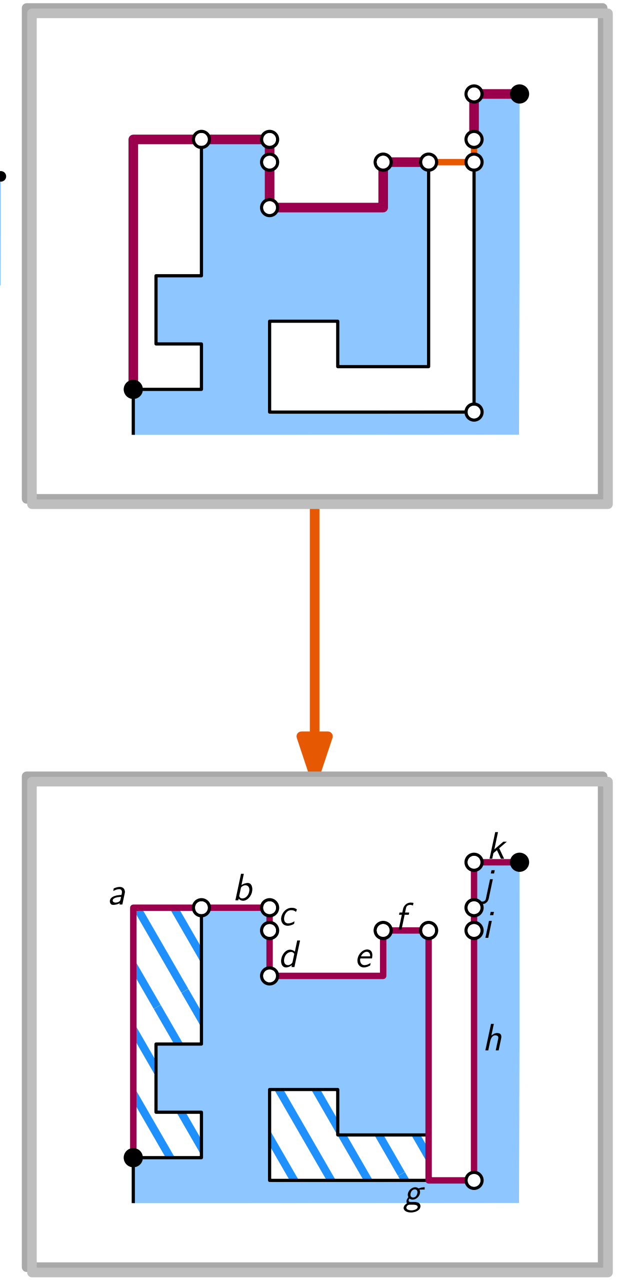
Decomposition tree of B_1.



Decomposition

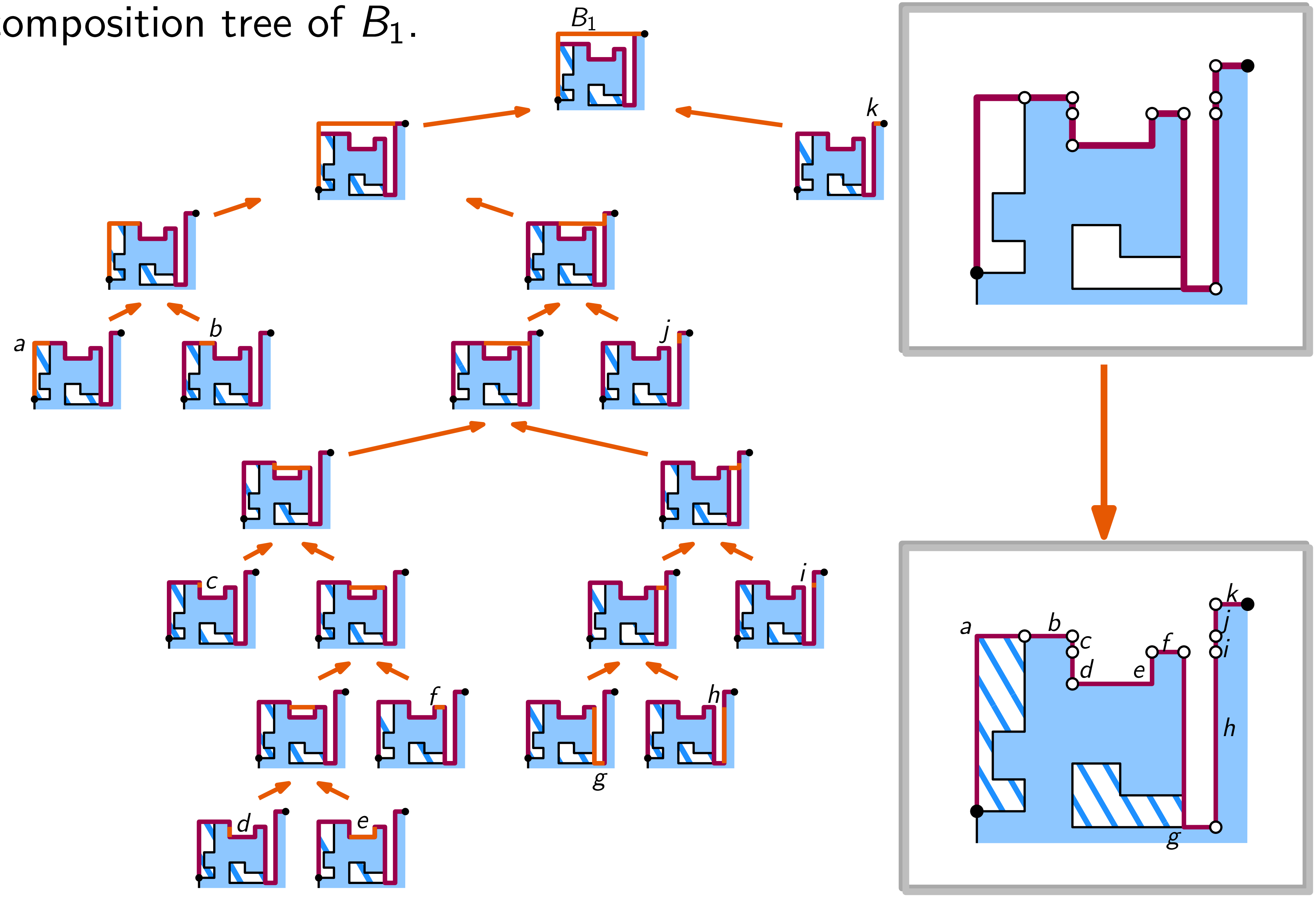
Decomposition tree of B_1.





Decomposition

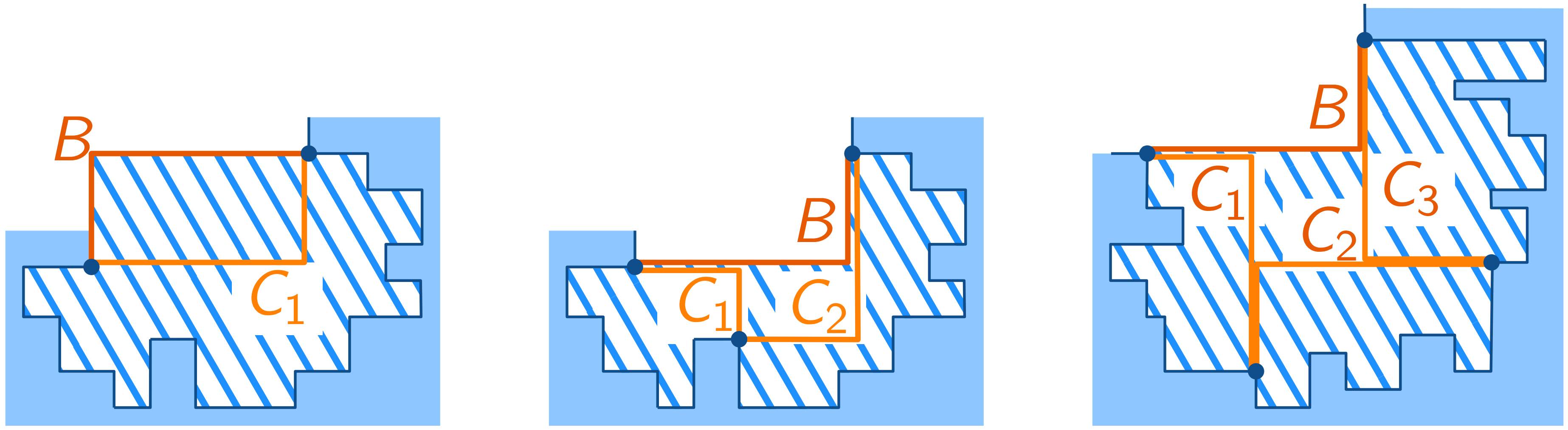
Decomposition tree of B_1.



Decomposition Rules

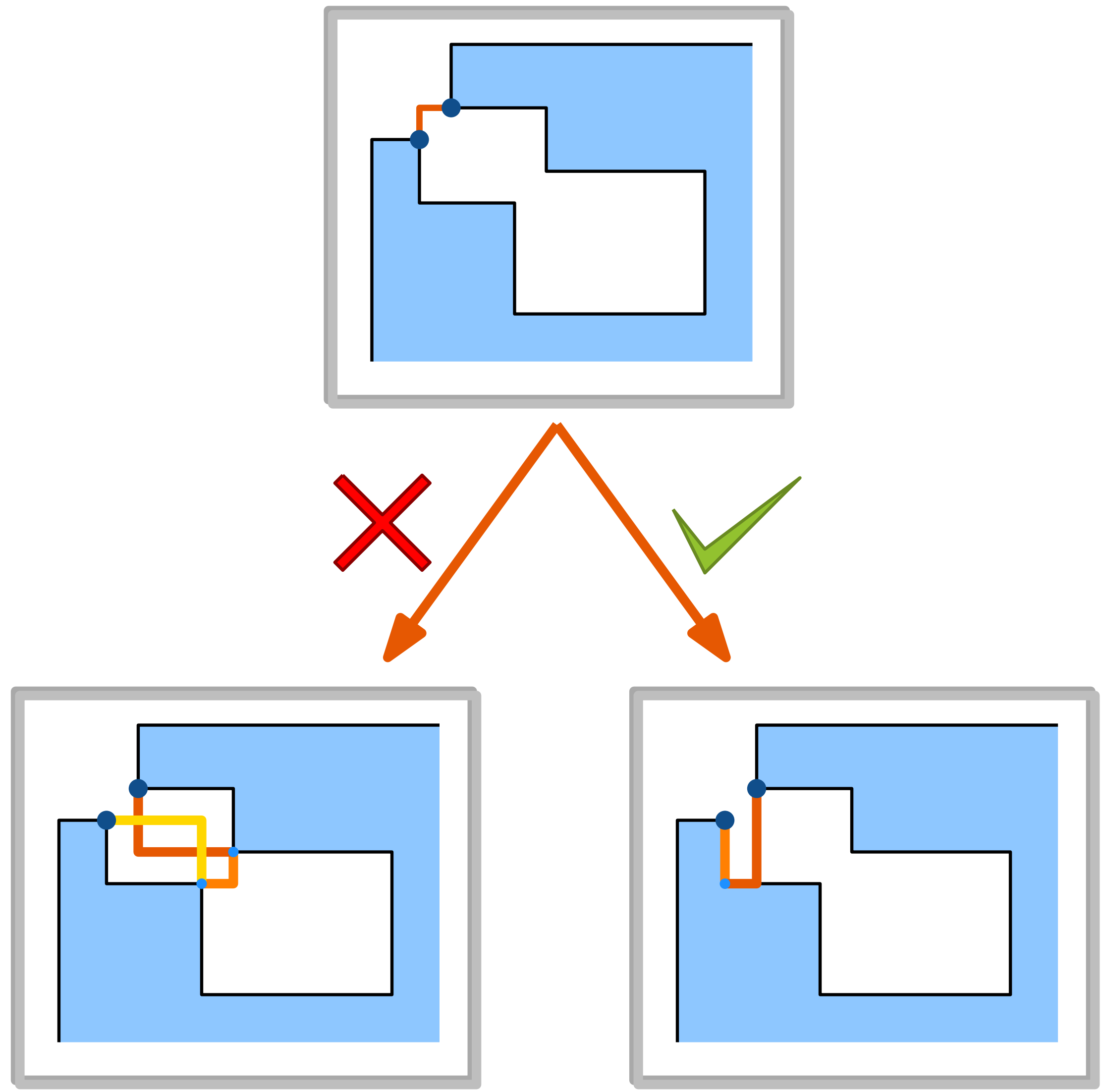
Decomposition of a bridge B into

- one to three connected bridges C_1, C_2, and C_3,
- each C_1, C_2, and C_3 lies in the bag of B
- the polyline defined by C_1, C_2, and C_3 connects the start and endpoint of B
- C_1, C_2, and C_3 may not cross each other pairwise



Decomposition Rules

Note:
rules guarantee that Q is not self-intersecting

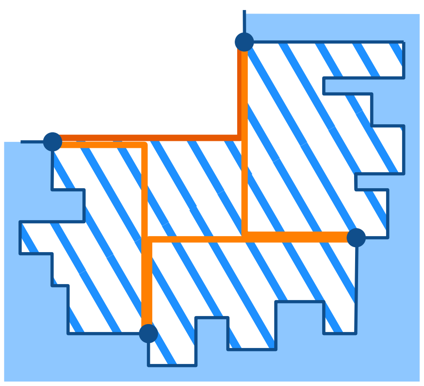


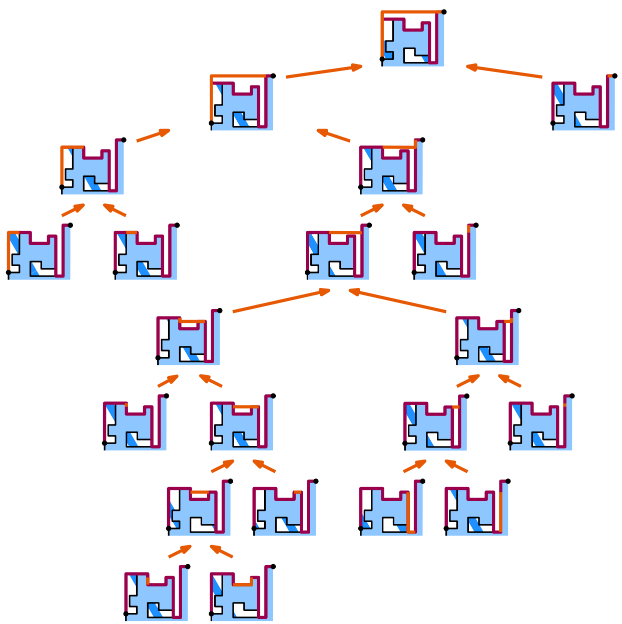
Computation

Lemma:
For each tight hull there exists a decomposition tree.

Observation:
Each decomposition of a bridge can be described by two additional points
$\Rightarrow$ all possible decompositions can be enumerated in polynomial time.

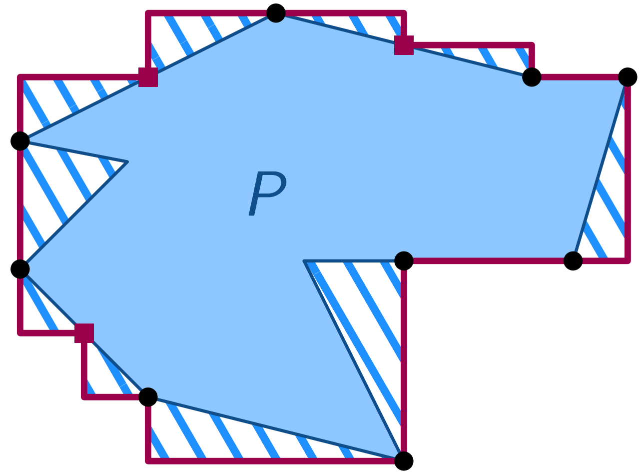
Use dynamic programming approach to build a decomposition tree of an α-optimal tight hull in $O(n^4)$ time.



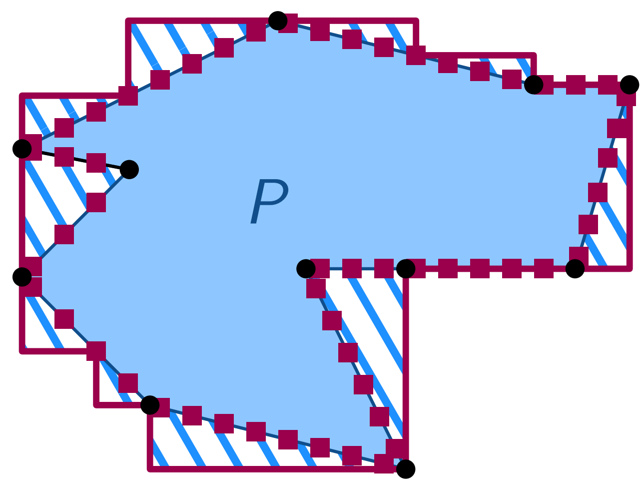


Non-Rectilinear Input Polygon

Problem:
vertices of Q are not necessarily vertices of the maximally subdivided P



Simple Approximative Approach:
sample regularly distibuted vertices on the edges of P



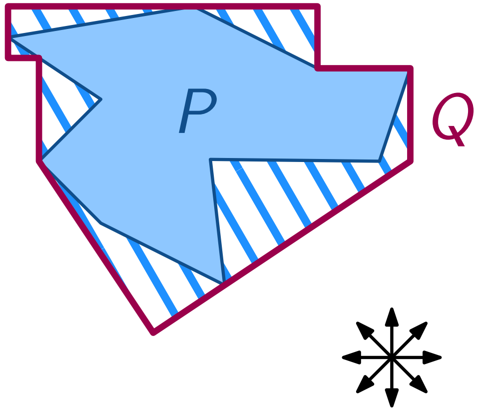
Conclusion and Outlook

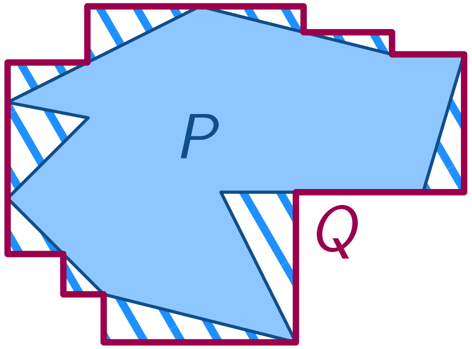
Conclusion:

- non-self intersecting α-optimal tight rectilinear hull in $O(n^4)$ time and $O(n^2)$ space

Future Work:

- $\mathcal{C}$-oriented tight hulls
- optimal solutions for arbitrary (simple) input polygons





Questions? Feedback?

bonerath@igg.uni-bonn.de
niedermann@igg.uni-bonn.de
haunert@igg.uni-bonn.de