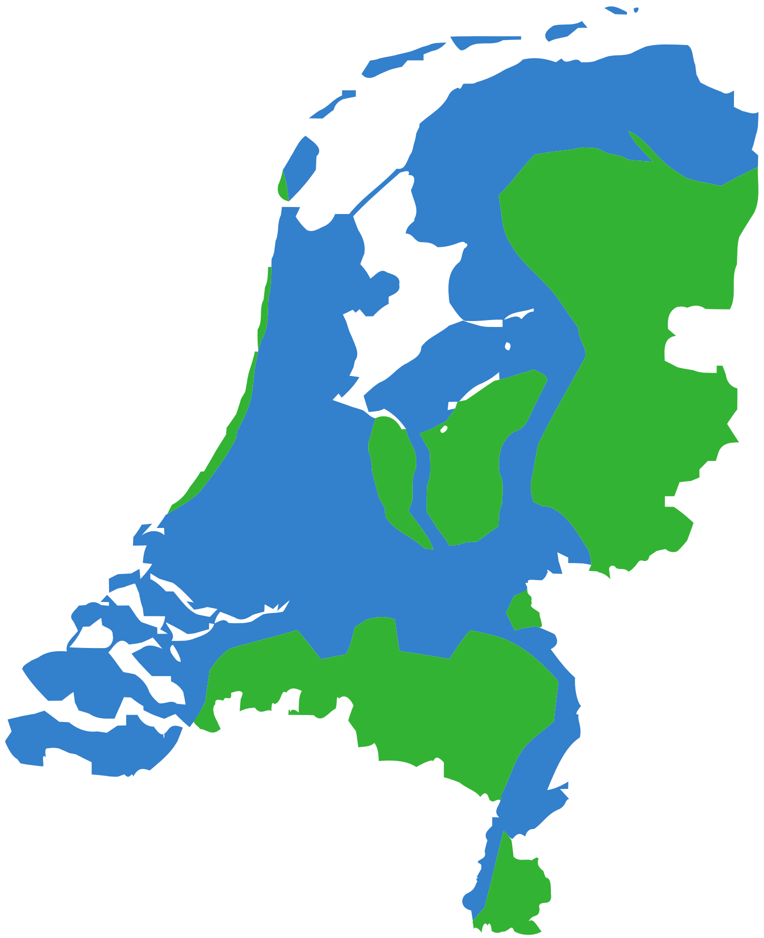
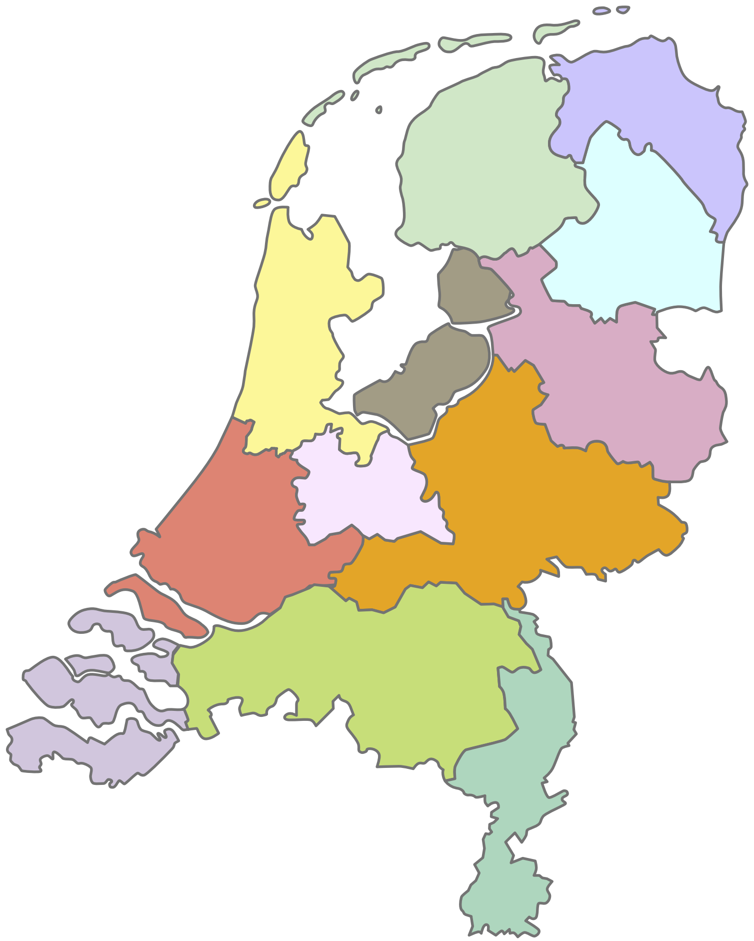
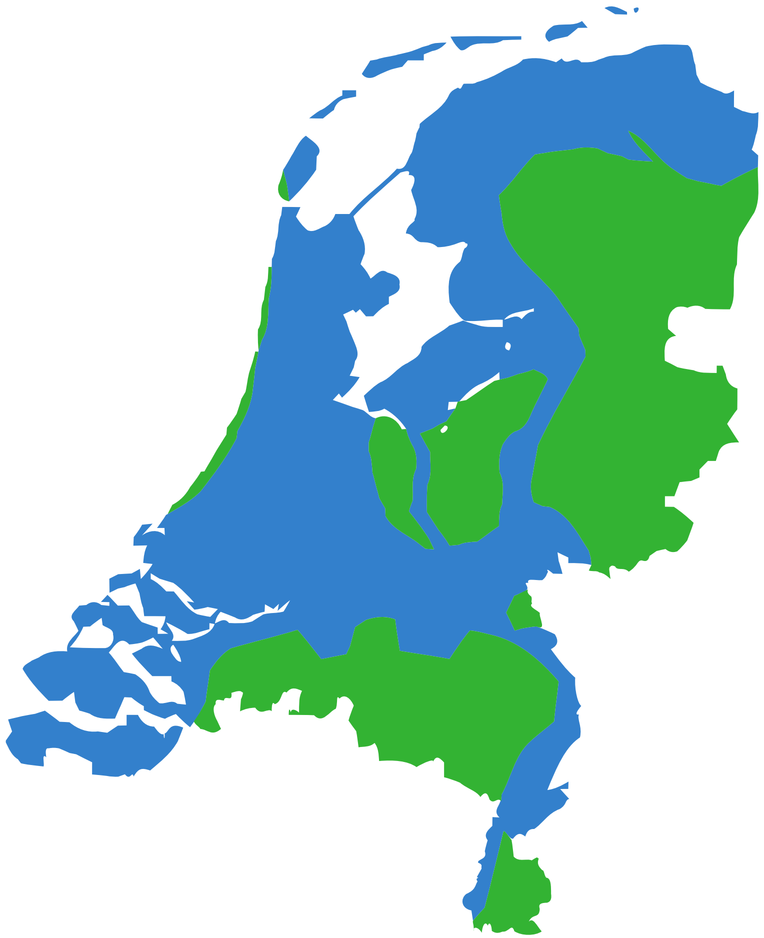
Introduction

Introduction



Introduction



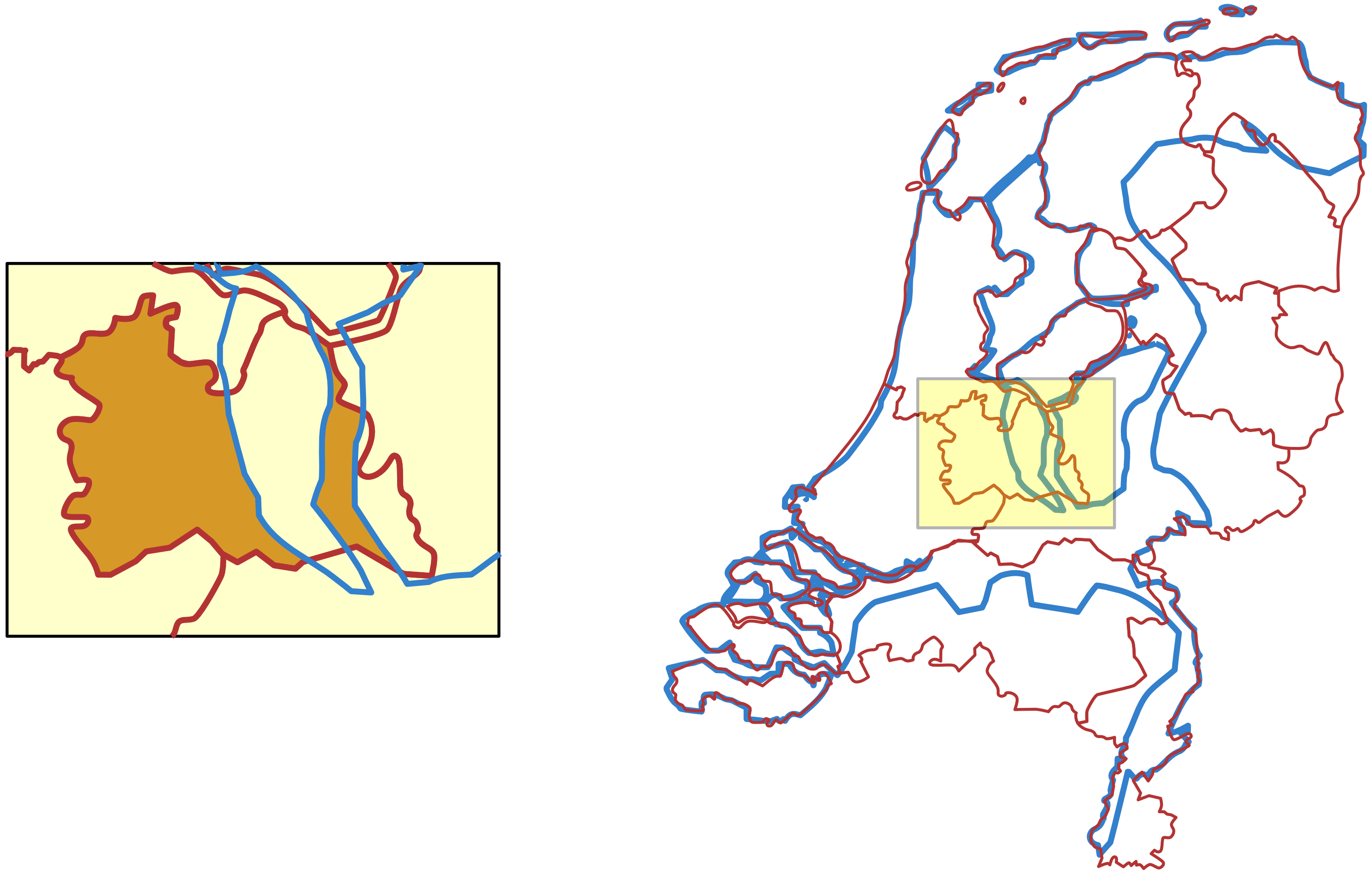


What percentage of Utrecht is in danger of floods?

Introduction

Introduction

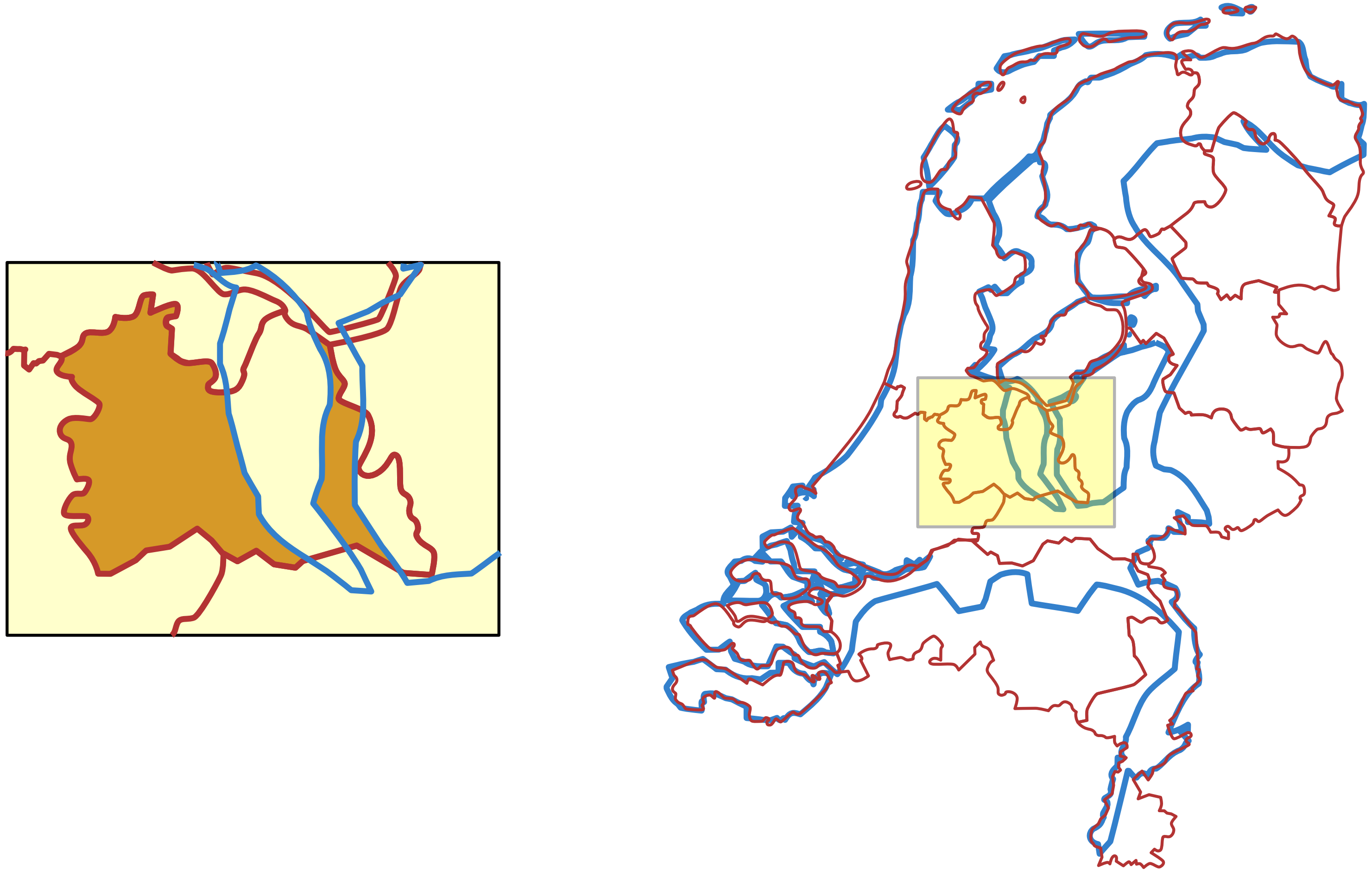
Introduction



Map Overlay

Problem:

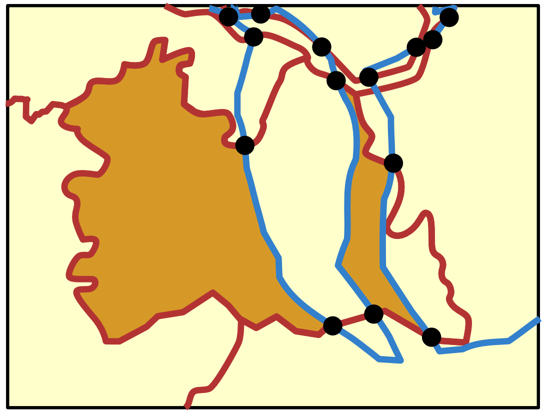
Given two maps M_1 and M_2, compute the overlay of M_1 and M_2.

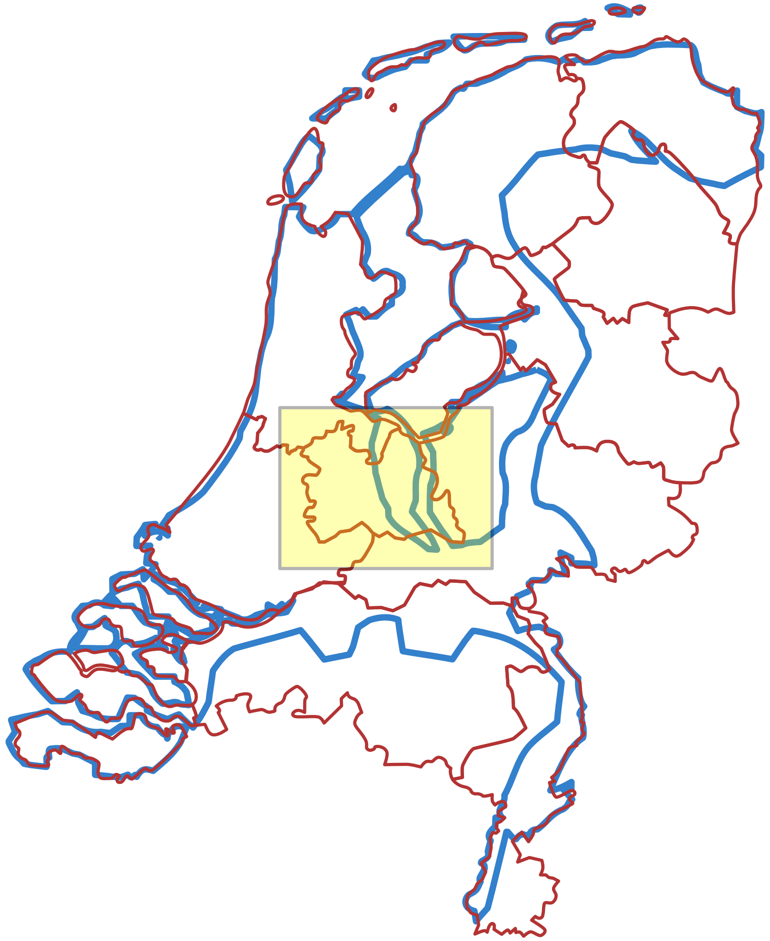


Map Overlay

Problem:

Given two maps M_1 and M_2, compute the overlay of M_1 and M_2.



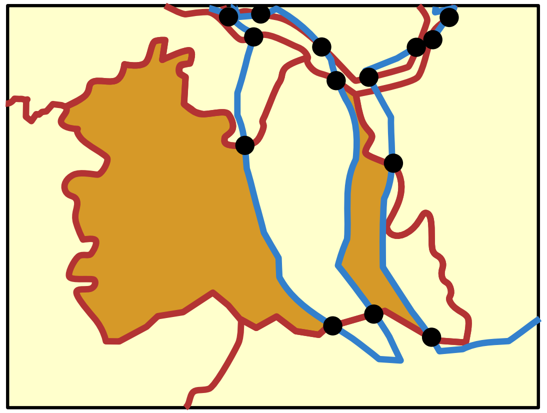


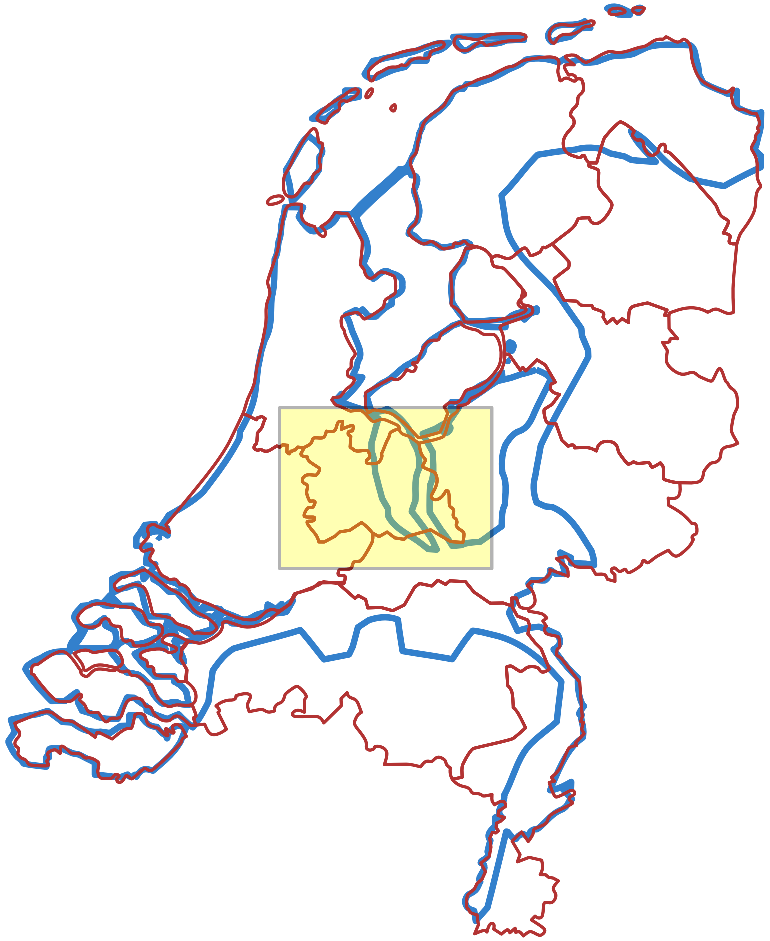
- How to compute all intersection points?

Map Overlay

Problem:

Given two maps M_1 and M_2, compute the overlay of M_1 and M_2.



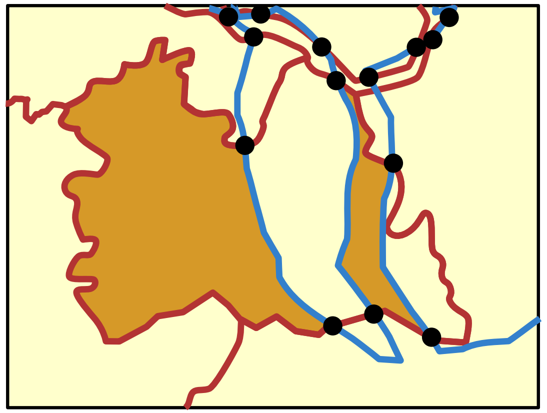


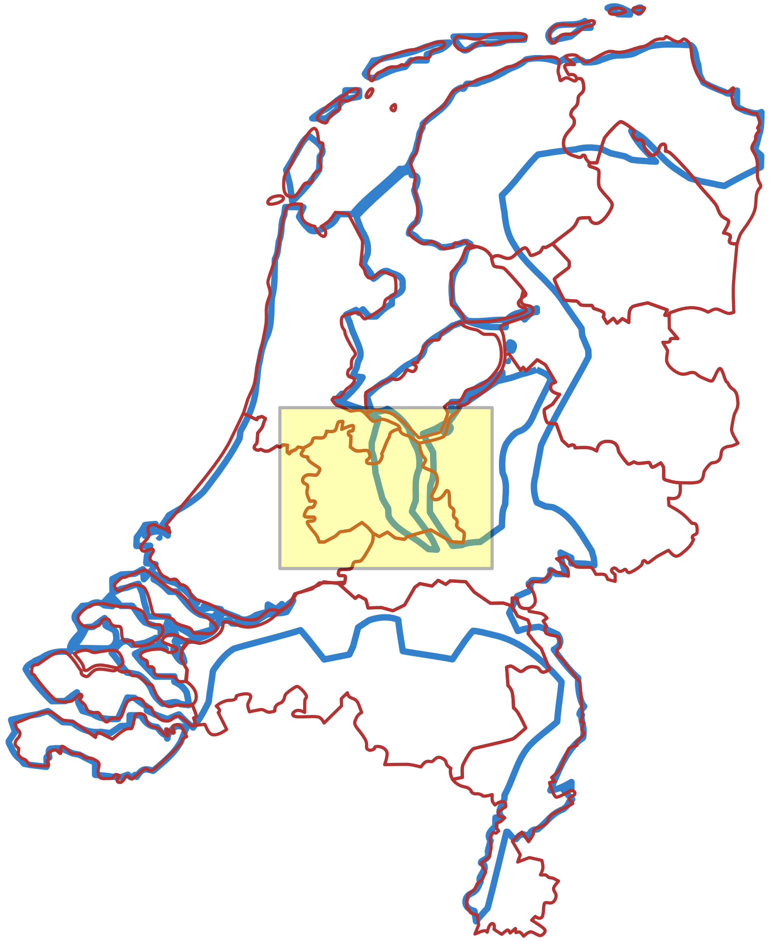
- How to compute all intersection points?
- How to represent (the overlay of) a map?

Map Overlay

Problem:

Given two maps M_1 and M_2, compute the overlay of M_1 and M_2.





- How to compute all intersection points?

- How to represent a planar subdivision?

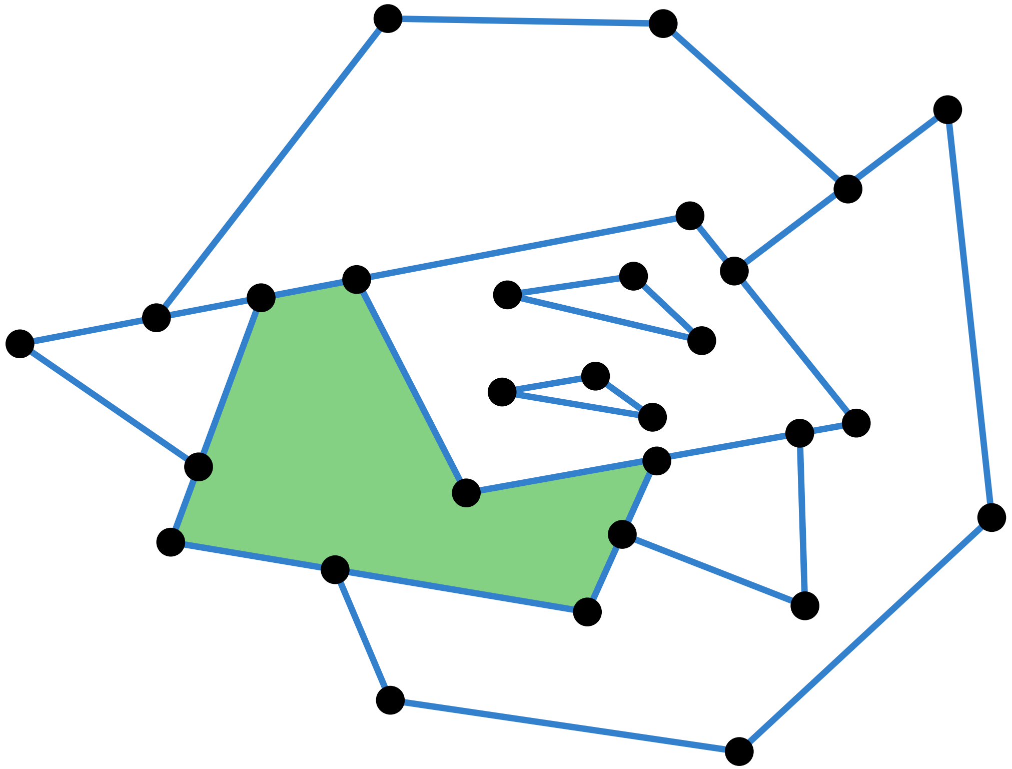
Representing a planar subdivision

A planar subdivision is an embedded planar graph, consisting of:

Vertices

Edges

Faces



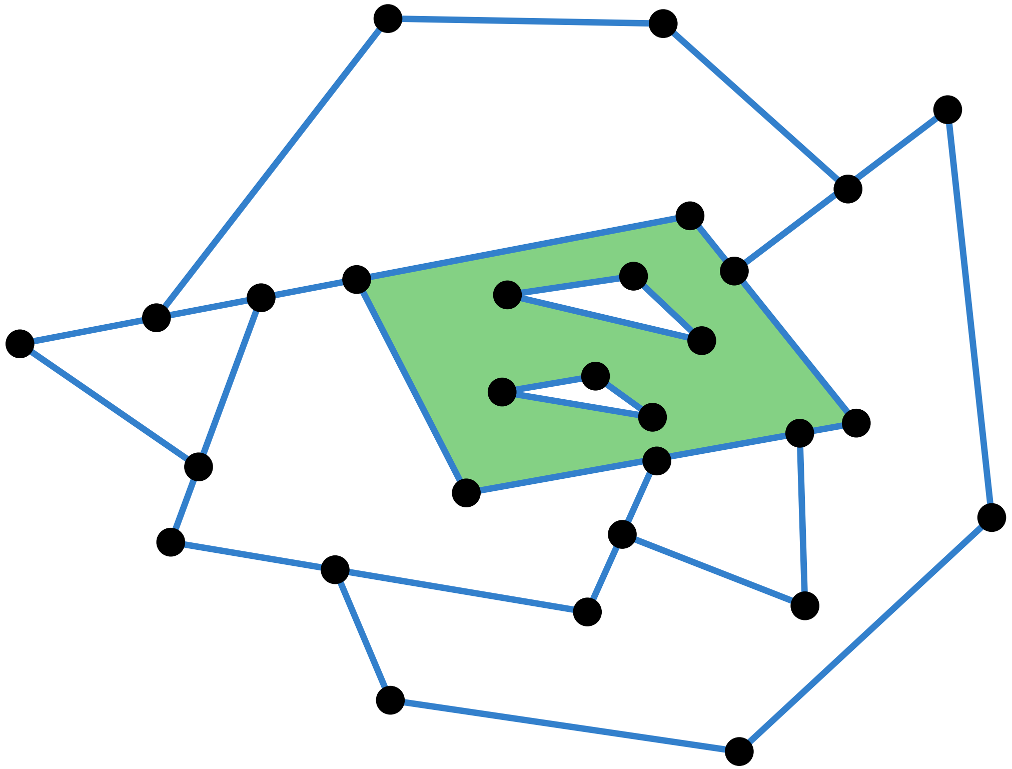
Representing a planar subdivision

A planar subdivision is an embedded planar graph, consisting of:

Vertices

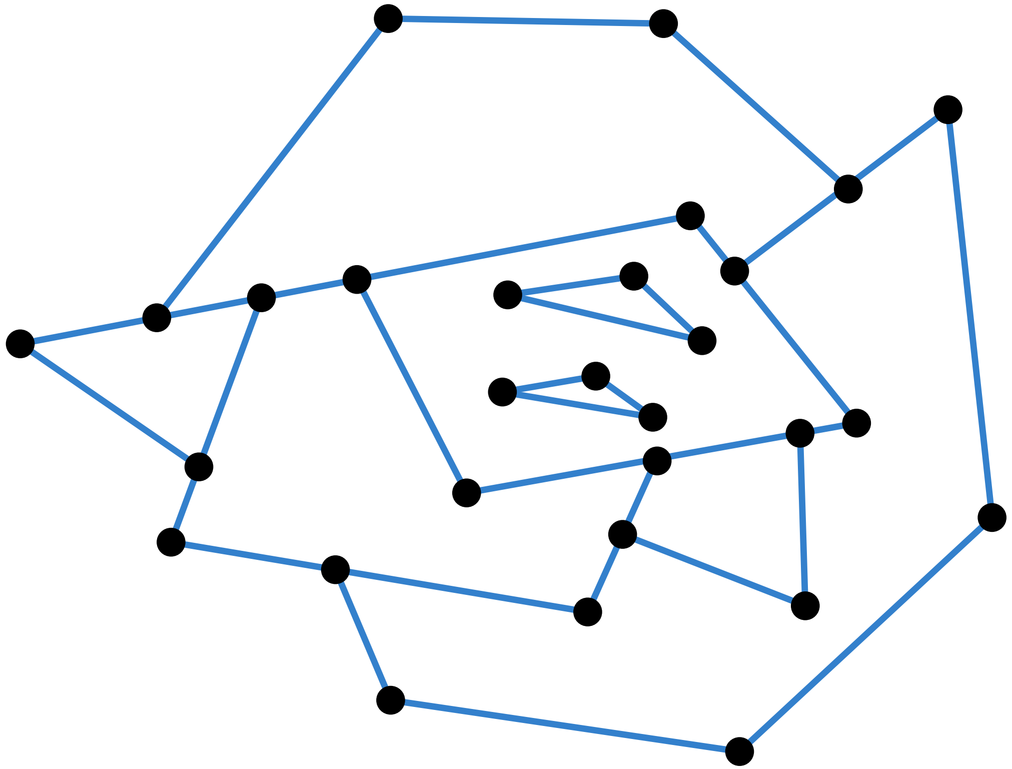
Edges

Faces



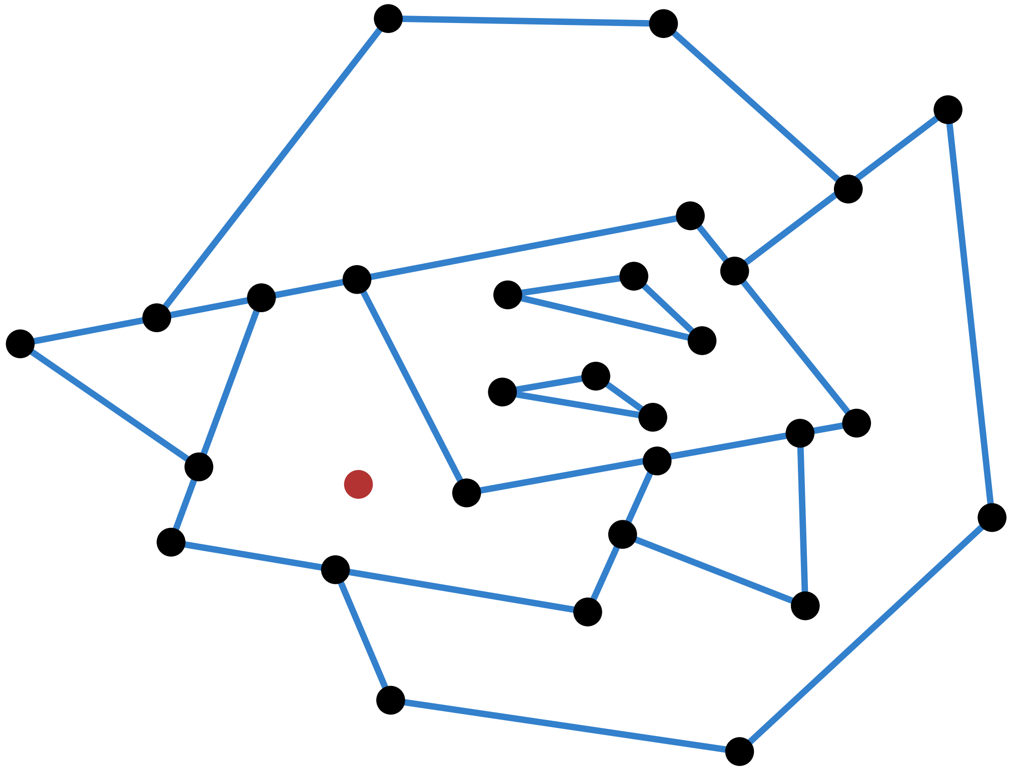
Representing a planar subdivision

What operations should are representation support?



Representing a planar subdivision

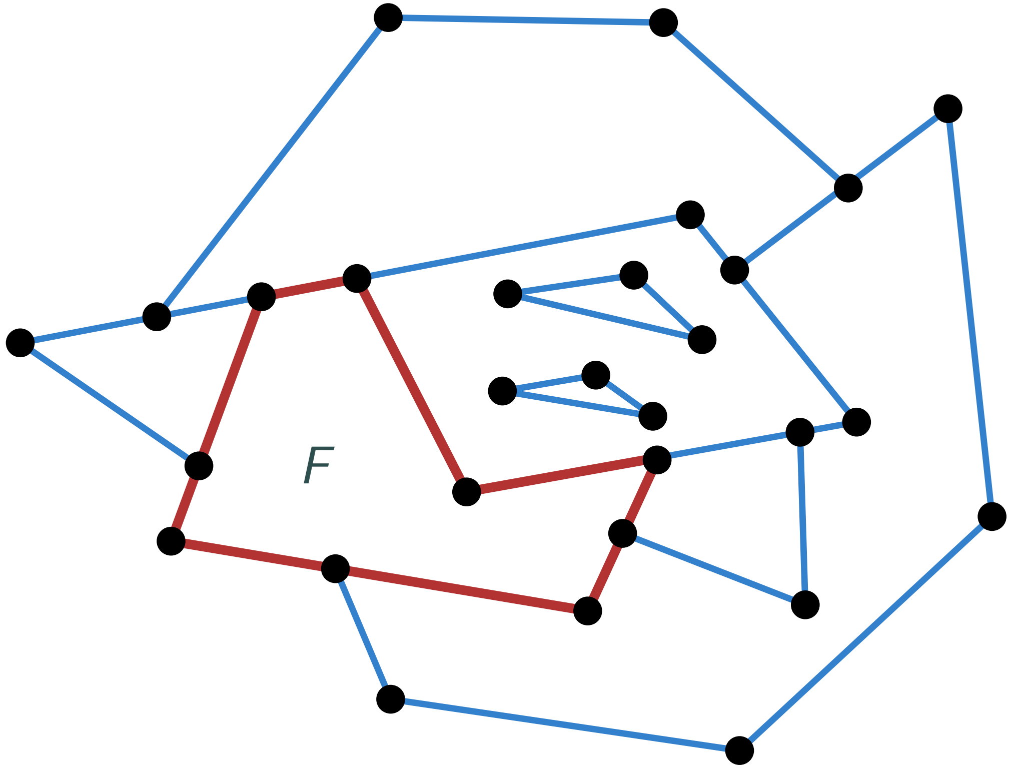
What operations should are representation support?



Representing a planar subdivision

What operations should are representation support?

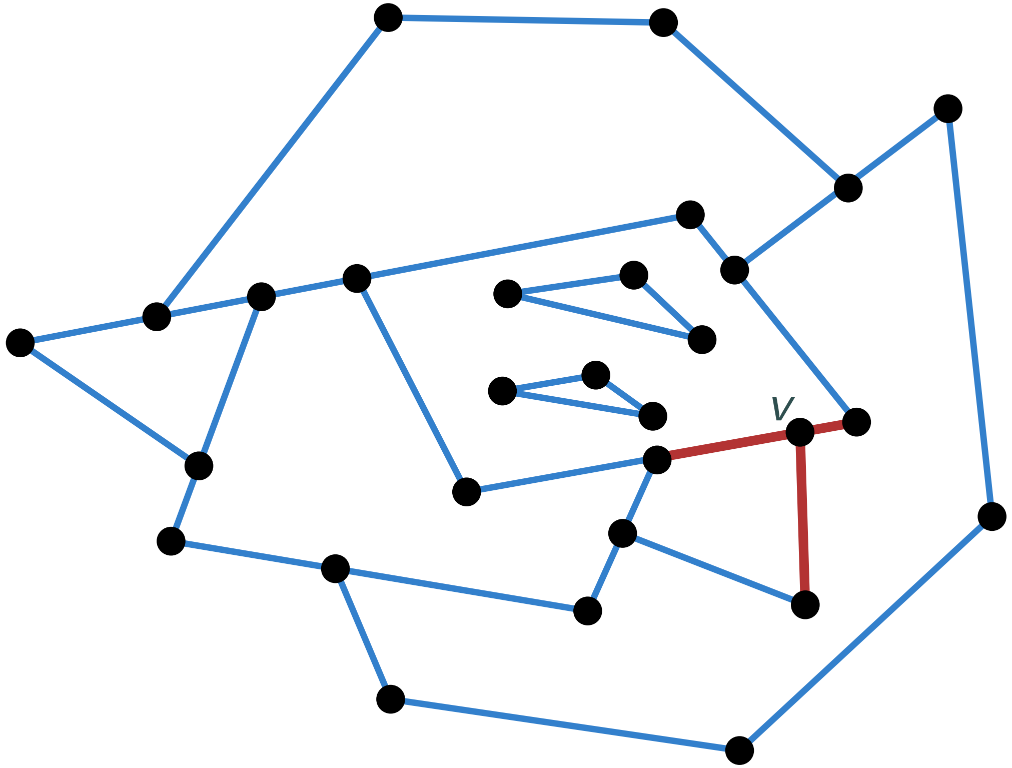
- List edges or vertices bounding a face F



Representing a planar subdivision

What operations should are representation support?

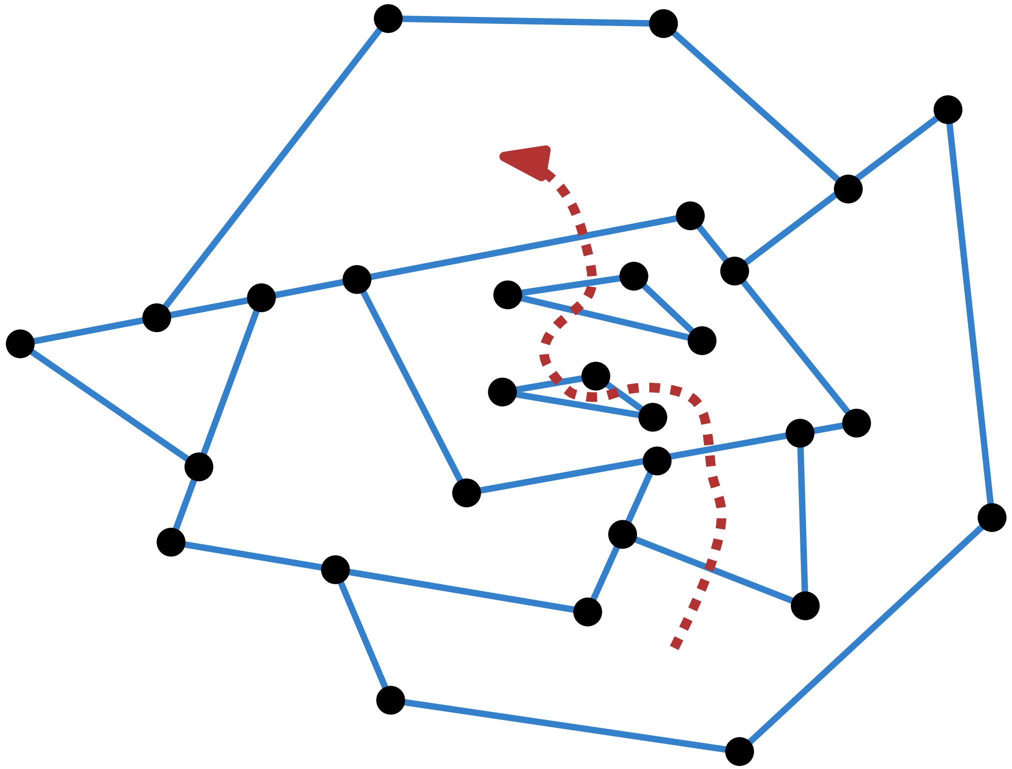
- List edges or vertices bounding a face F
- List edges incident to a vertex v



Representing a planar subdivision

What operations should are representation support?

- List edges or vertices bounding a face F
- List edges incident to a vertex v
- Efficiently "walk" in the subdivision
- ...



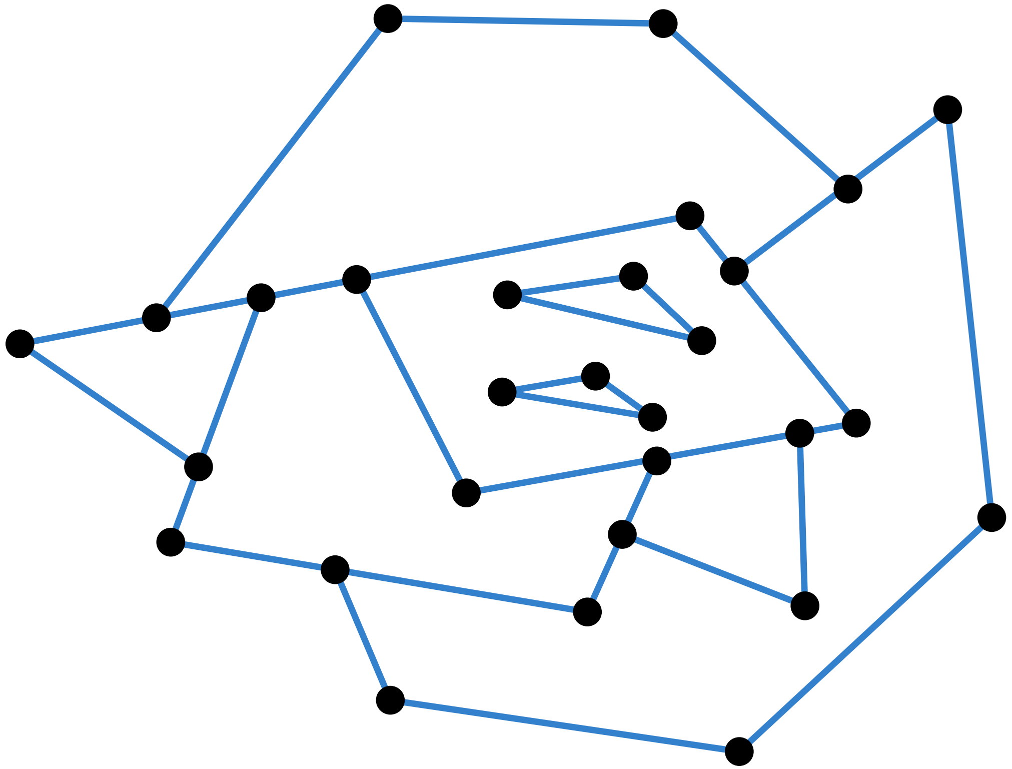
The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a doubly-connected edge list (DCEL)

Vertices

Edges

Faces



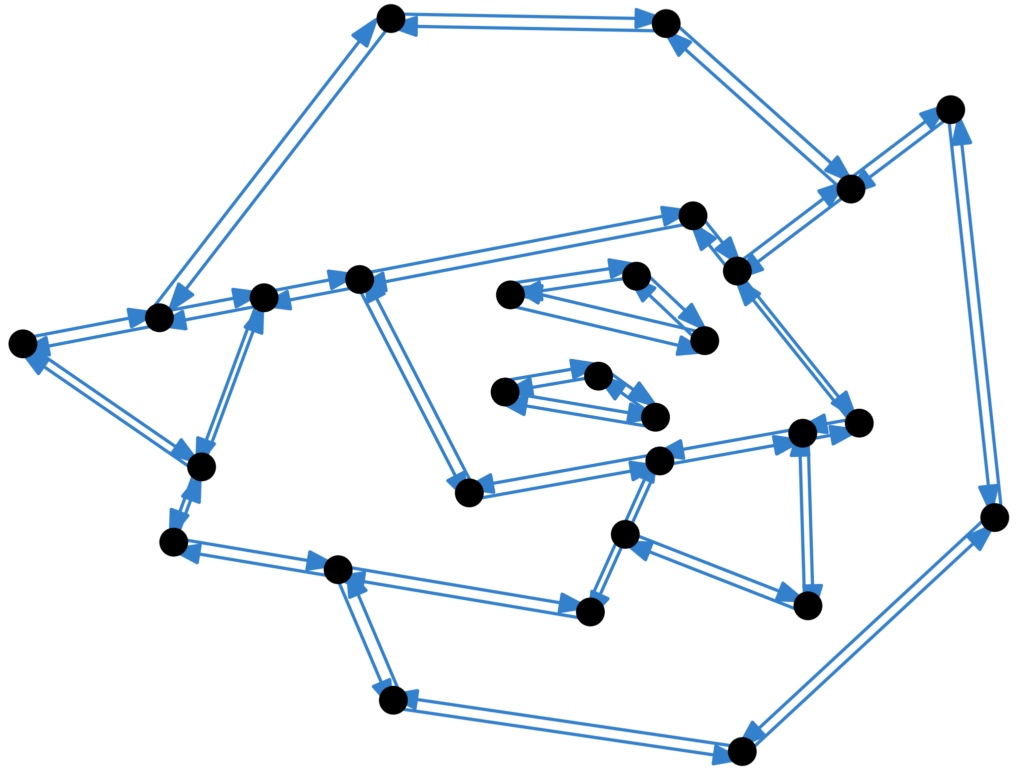
The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a doubly-connected edge list (DCEL)

Vertices

Half-edges

Faces



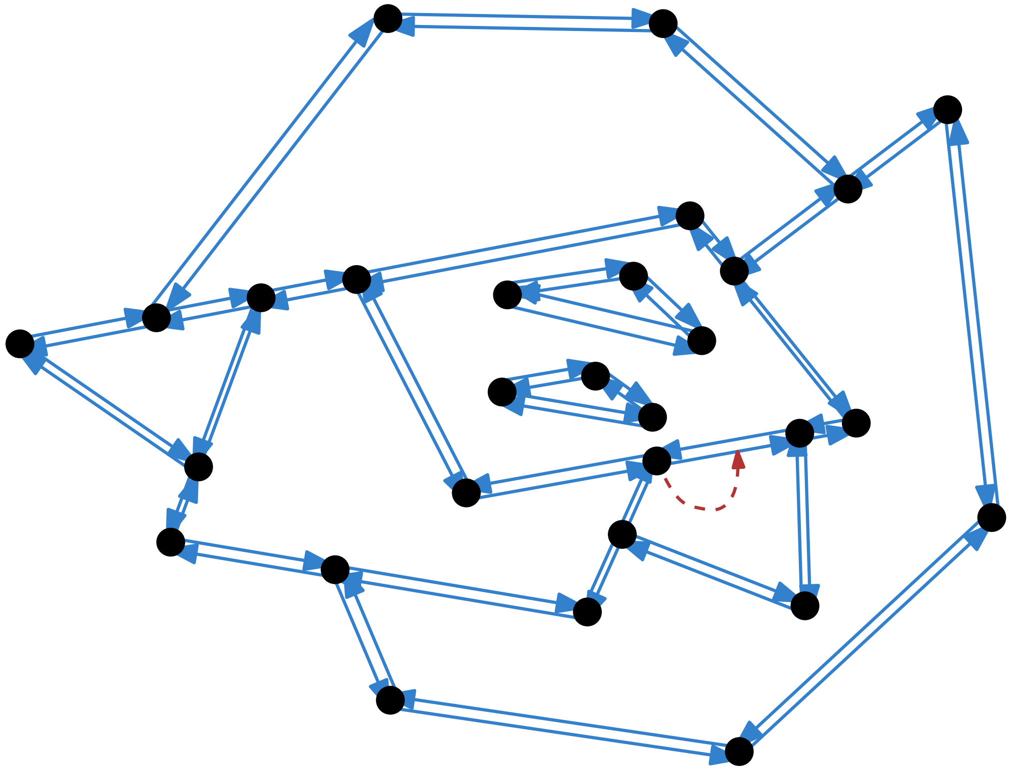
The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a
doubly-connected edge list (DCEL)

Vertices

Coordinates

An arbitrary outgoing half-edge

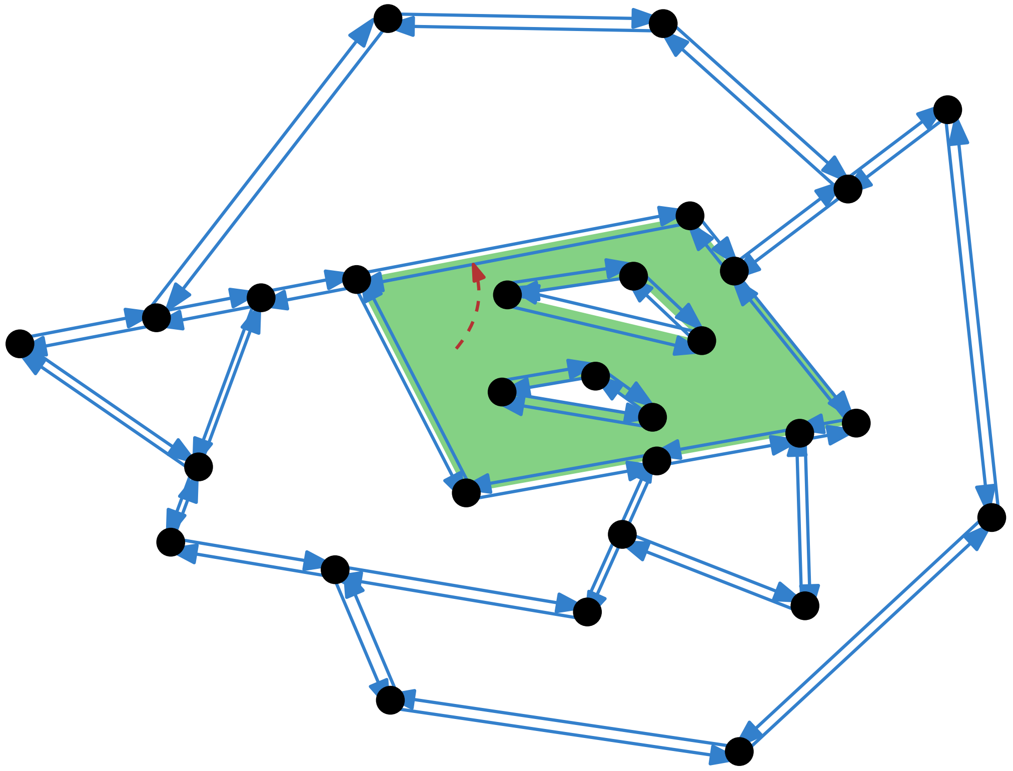


Half-edges

Faces

The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a
doubly-connected edge list (DCEL)



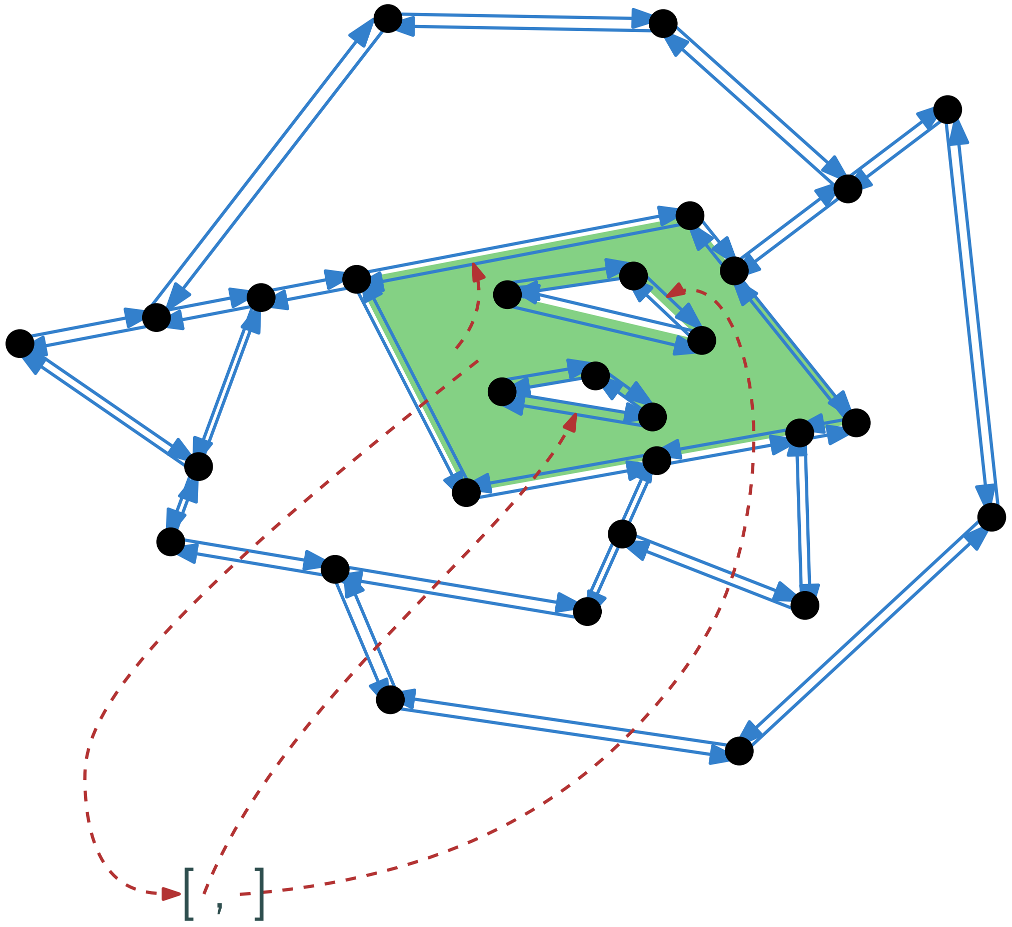
Vertices
- Coordinates
- An arbitrary outgoing half-edge

Half-edges

Faces
- An arbitrary half-edge on its outer boundary

The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a
doubly-connected edge list (DCEL)



Vertices

Coordinates

An arbitrary outgoing half-edge

Half-edges

Faces

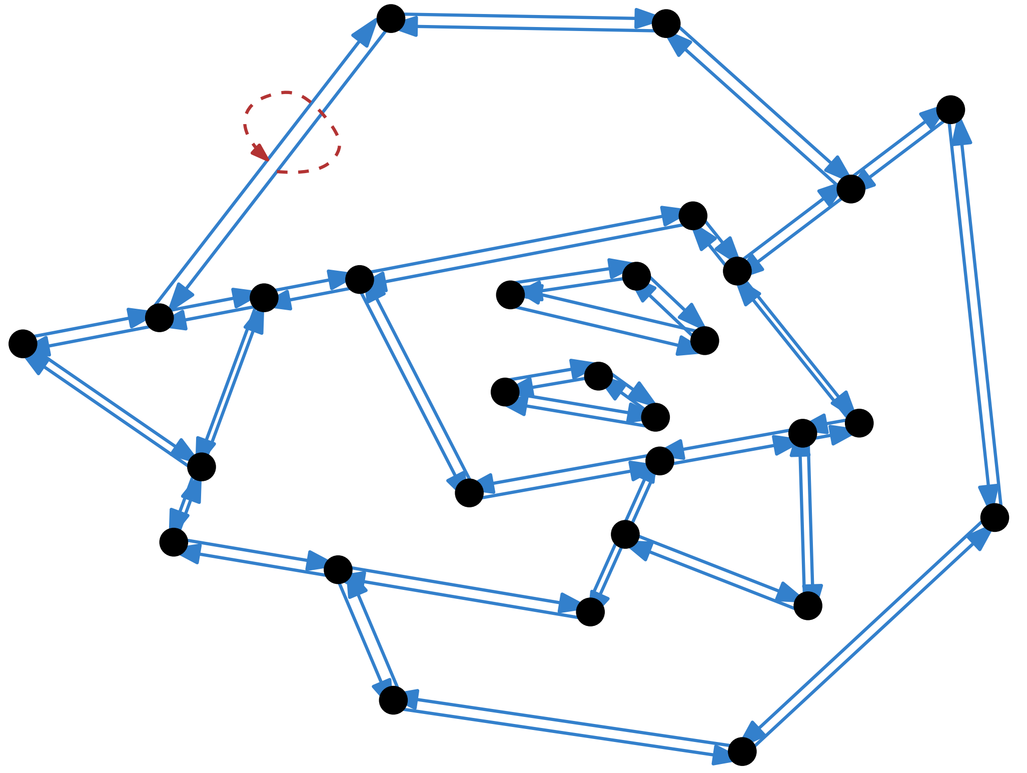
An arbitrary half-edge on its outer boundary

A list of holes in the face;

for each hole an arbitrary half-edge.

The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a doubly-connected edge list (DCEL)



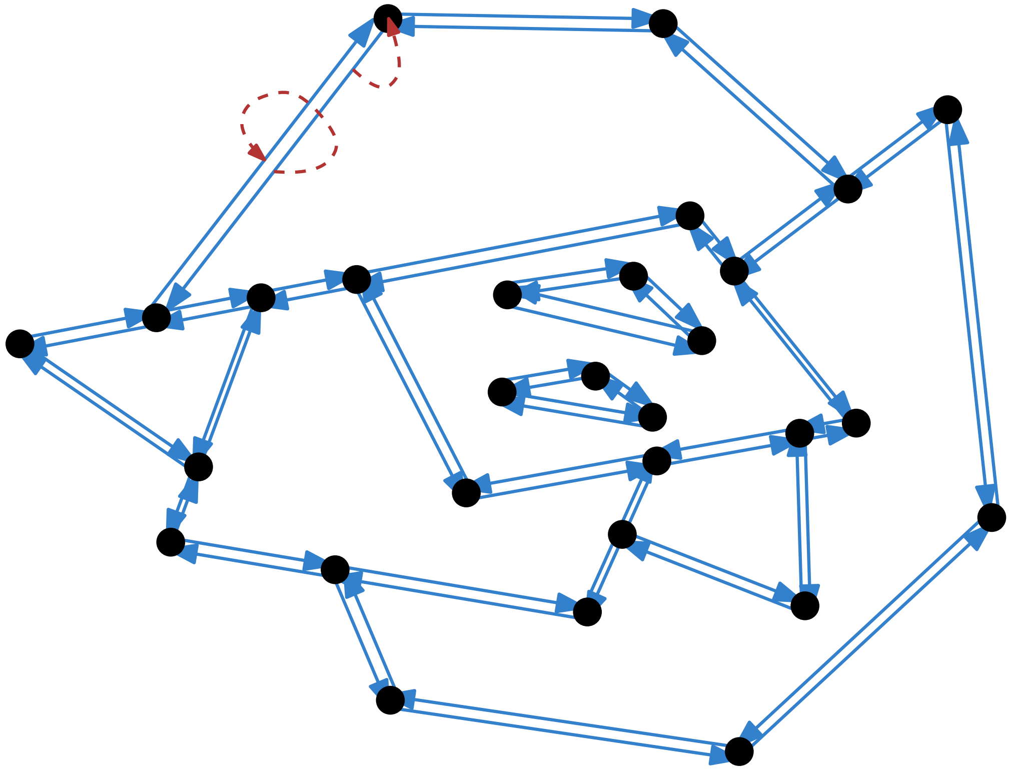
Vertices
- Coordinates
- An arbitrary outgoing half-edge

Half-edges
- Twin

Faces
- An arbitrary half-edge on its outer boundary
- A list of holes in the face;
- for each hole an arbitrary half-edge.

The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a doubly-connected edge list (DCEL)



Vertices

Coordinates

An arbitrary outgoing half-edge

Half-edges

Twin

Origin

Faces

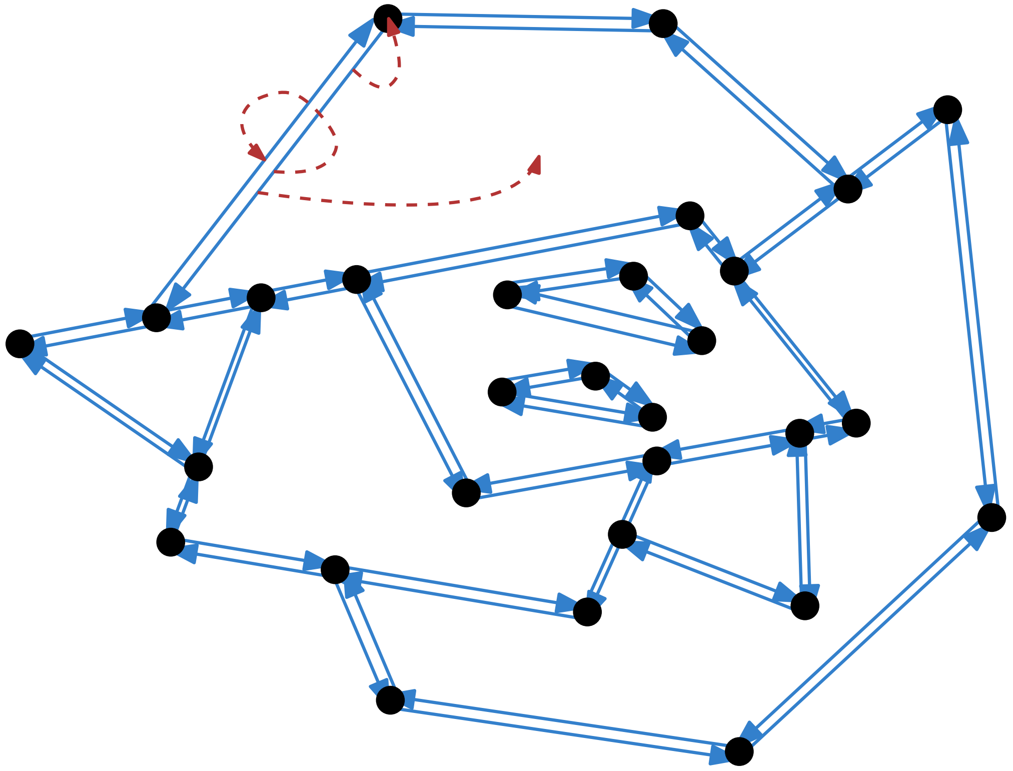
An arbitrary half-edge on its outer boundary

A list of holes in the face;

for each hole an arbitrary half-edge.

The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a
doubly-connected edge list (DCEL)



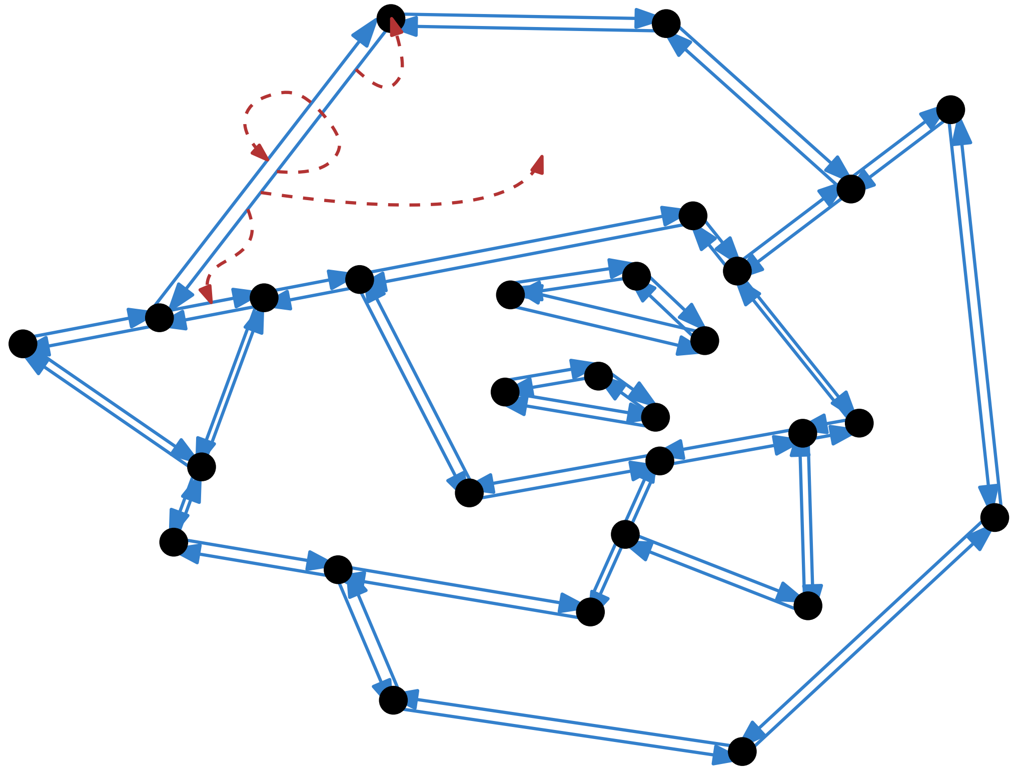
Vertices
- Coordinates
- An arbitrary outgoing half-edge

Half-edges
- Twin
- Origin
- Incident face

Faces
- An arbitrary half-edge on its outer boundary
- A list of holes in the face;
- for each hole an arbitrary half-edge.

The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a doubly-connected edge list (DCEL)



Vertices

Coordinates

An arbitrary outgoing half-edge

Half-edges

Twin

Origin

Incident face

Next

Faces

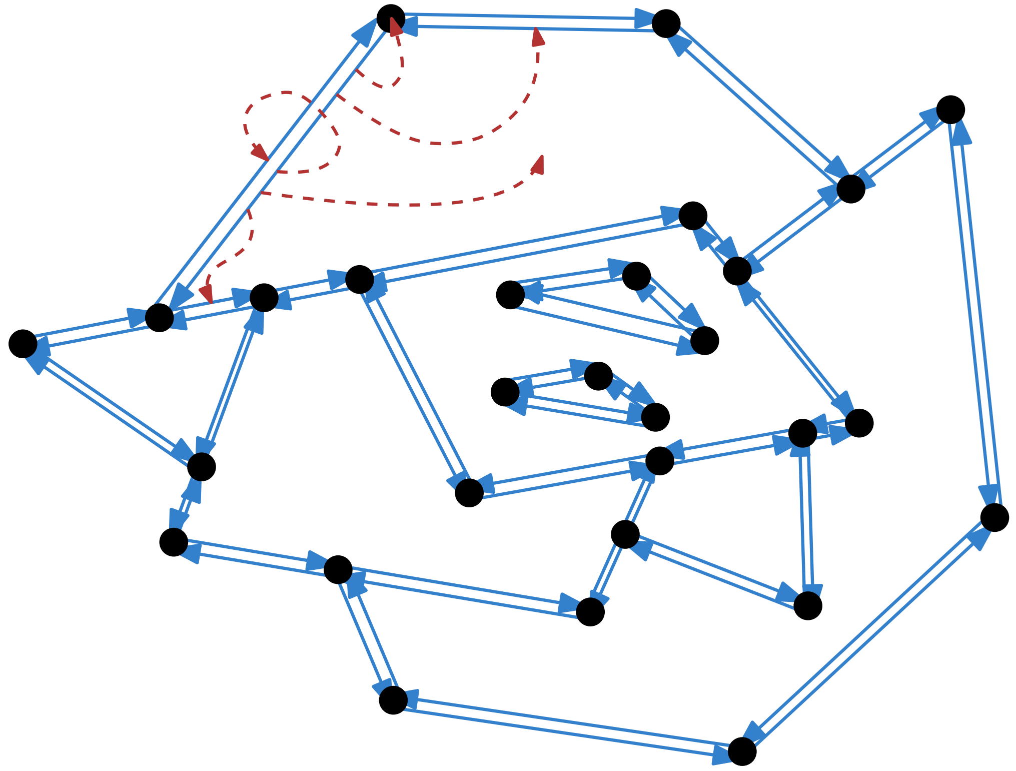
An arbitrary half-edge on its outer boundary

A list of holes in the face;

for each hole an arbitrary half-edge.

The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a doubly-connected edge list (DCEL)



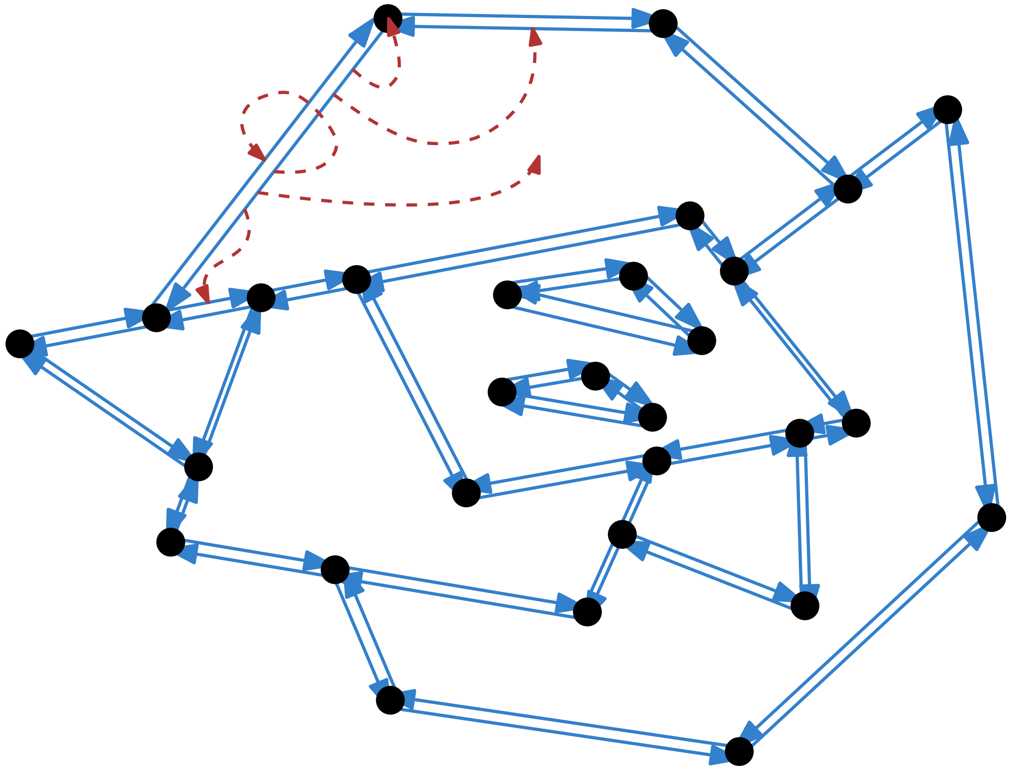
Vertices
- Coordinates
- An arbitrary outgoing half-edge

Half-edges
- Twin
- Origin
- Incident face
- Next
- Prev

Faces
- An arbitrary half-edge on its outer boundary
- A list of holes in the face;
- for each hole an arbitrary half-edge.

The Doubly-Connected Edge List (DCEL)

We store the planar subdivision in a doubly-connected edge list (DCEL)



Vertices
- Coordinates
- An arbitrary outgoing half-edge
- Label

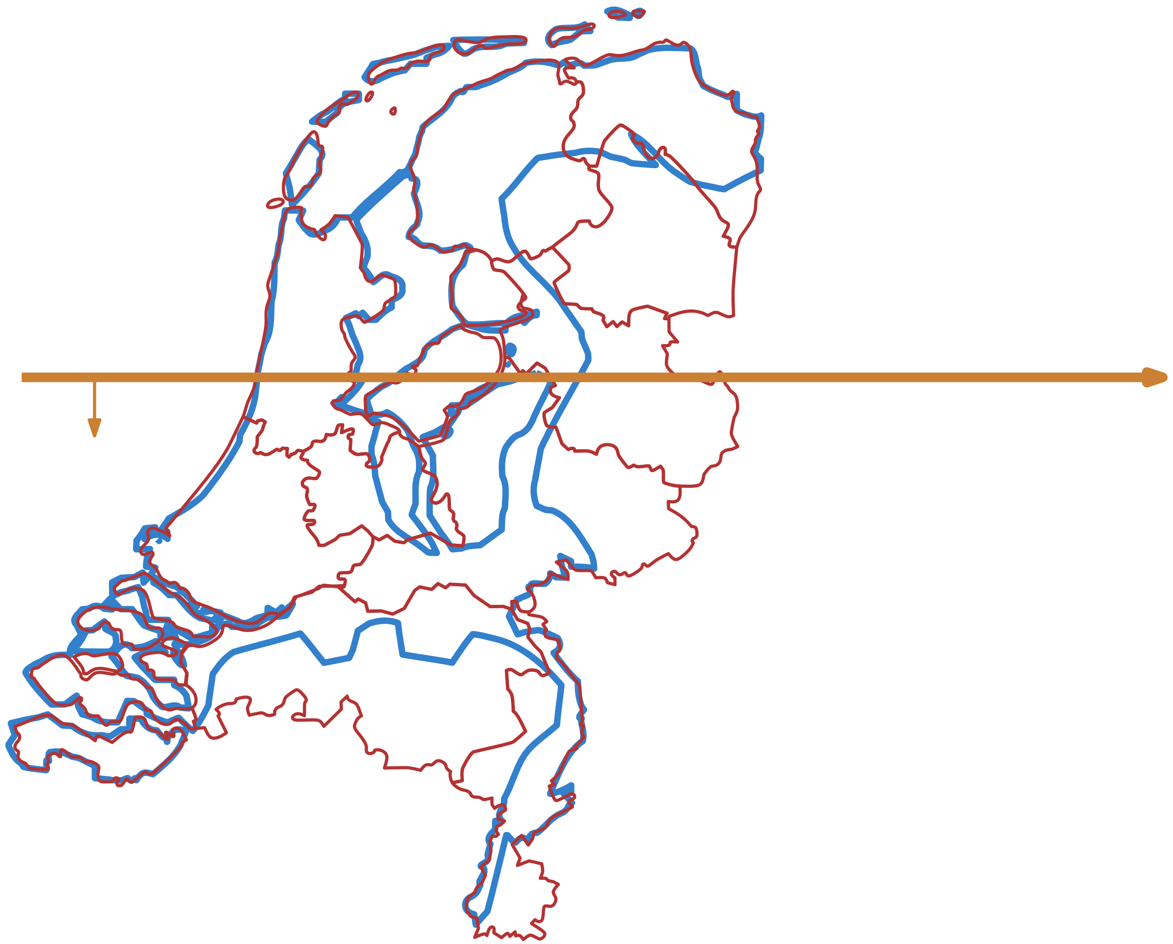
Half-edges
- Twin
- Origin
- Incident face
- Next
- Prev
- Label

Faces
- An arbitrary half-edge on its outer boundary
- A list of holes in the face;
- for each hole an arbitrary half-edge.
- Label

Map Overlay

Compute the overlay of M_1 and M_2:

Use a sweep line algorithm to compute the intersection points and maintain the vertex and half-edge information above the sweep line.

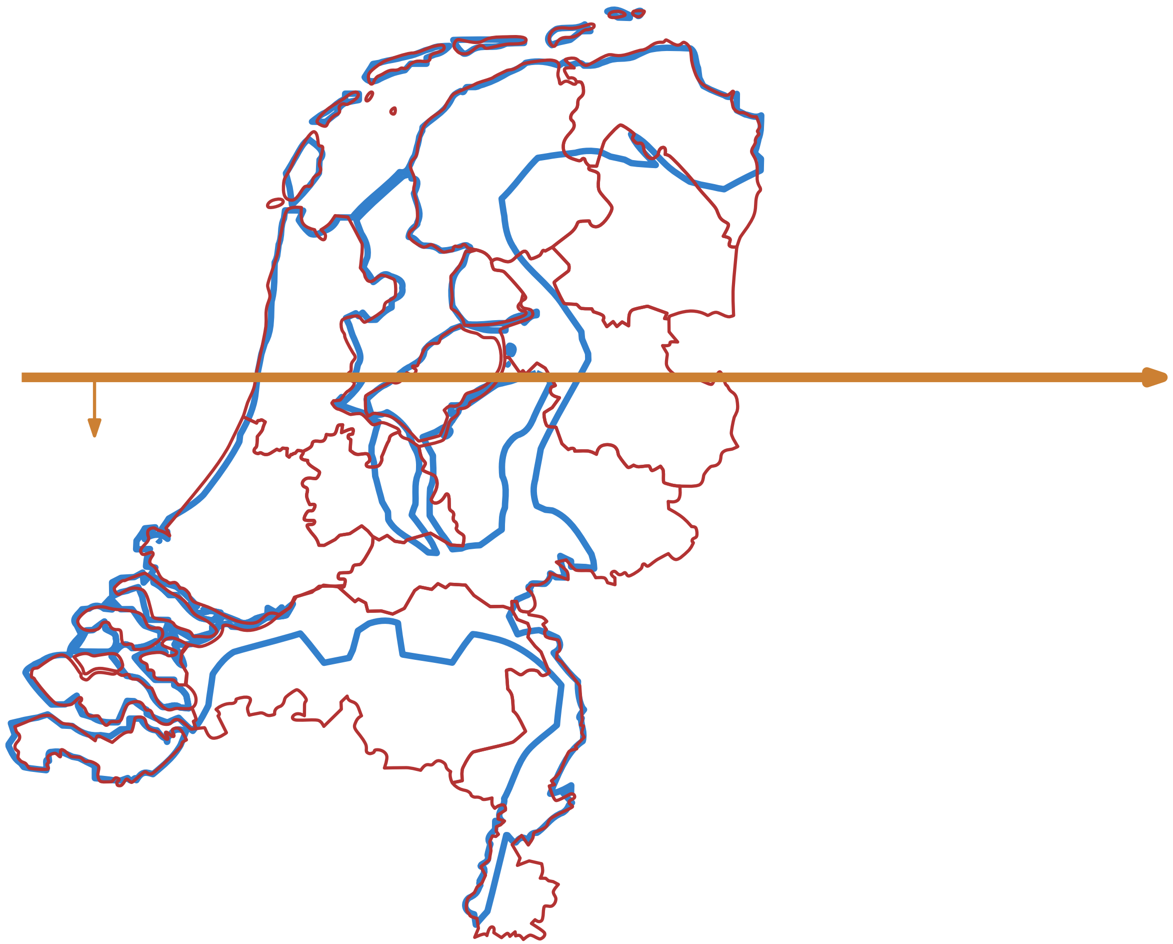


Map Overlay

Compute the overlay of M_1 and M_2:

Use a sweep line algorithm to compute the intersection points and maintain the vertex and half-edge information above the sweep line.

Use a second sweep to compute the face information.

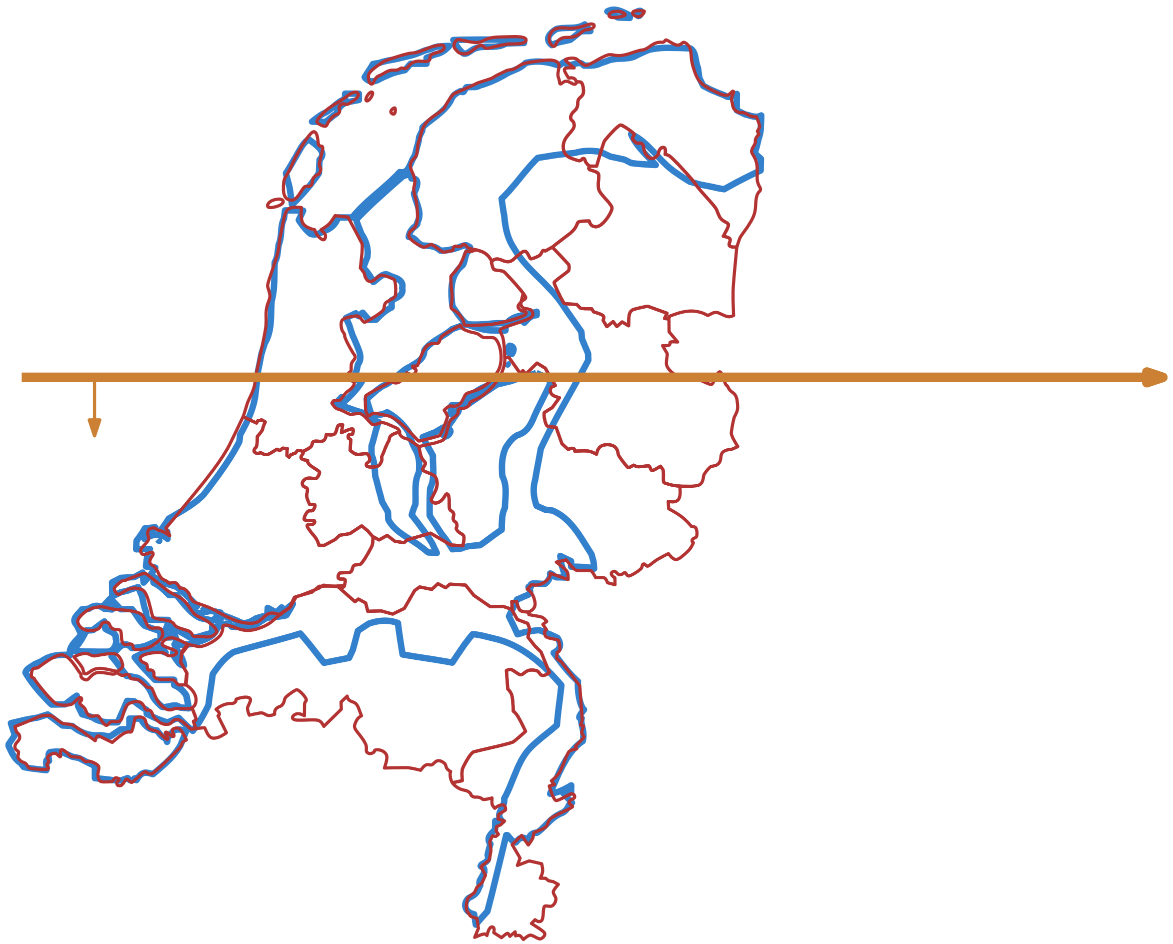


Map Overlay

Compute the overlay of M_1 and M_2:

Use a sweep line algorithm to compute the intersection points and maintain the vertex and half-edge information above the sweep line.

Use a second sweep to compute the face information.



Theorem. We can compute the overlay of M_1 and M_2 in $O((n+k)\log n)$ time, where n is the total number of line segments in M_1 and M_2, and k is the number of line segments in the output.