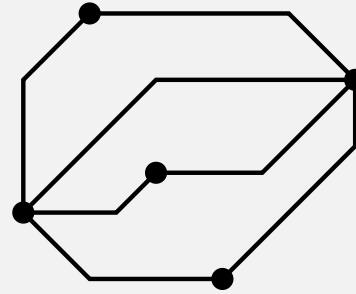
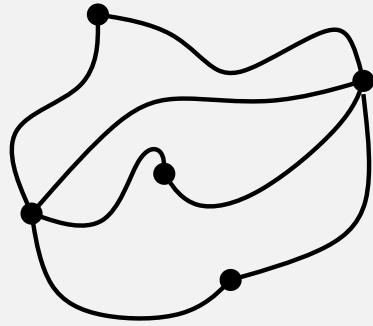


Schematization of Road Networks

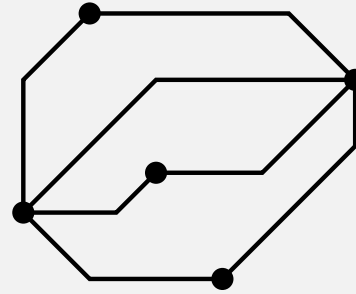
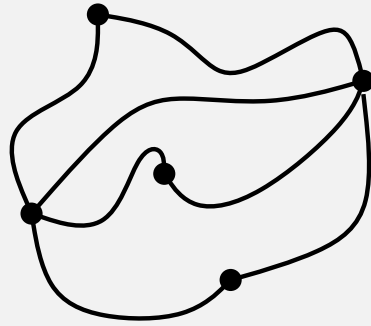
S. Cabello, M. de Berg, S. van Dijk,
M. van Kreveld, T. Strijk



What is a schematic map?



What is a schematic map?

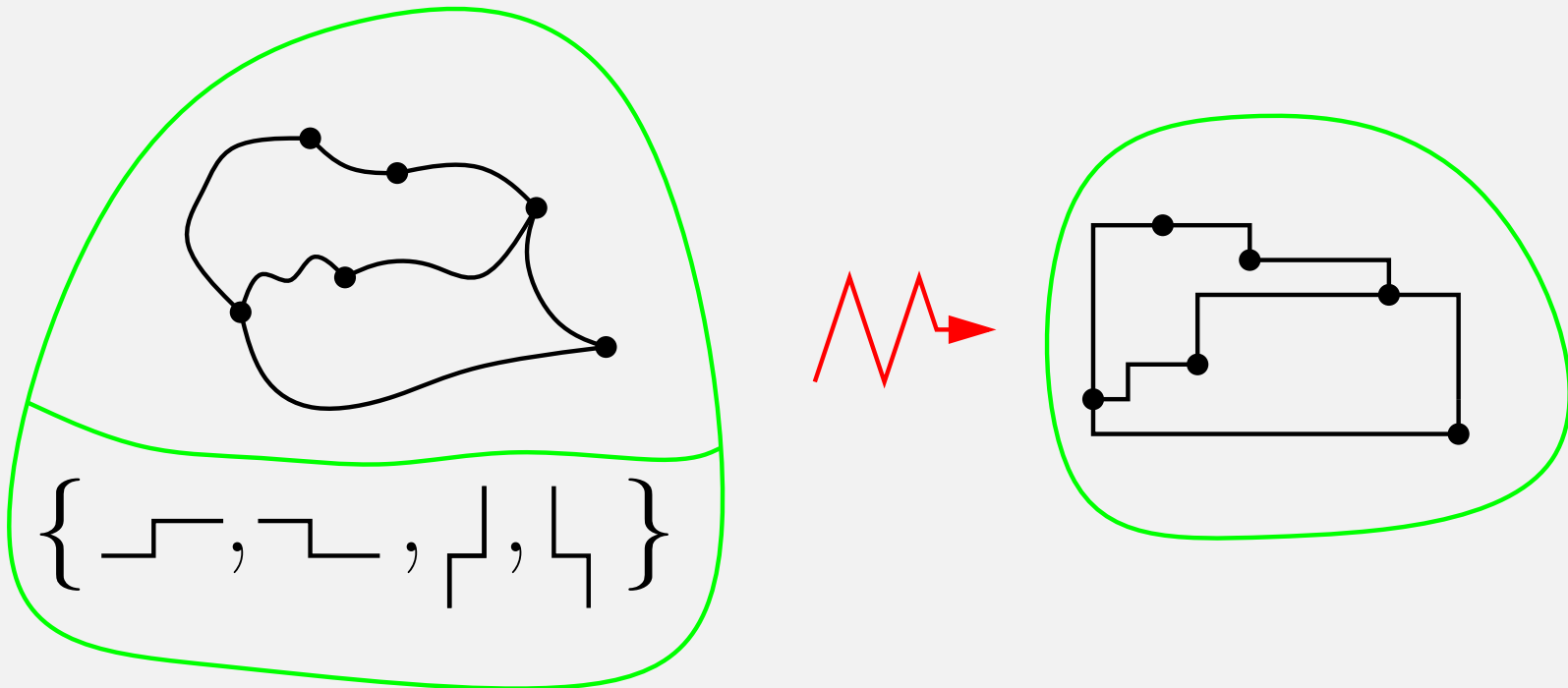


Drastic simplification and conceptualisation of connections.

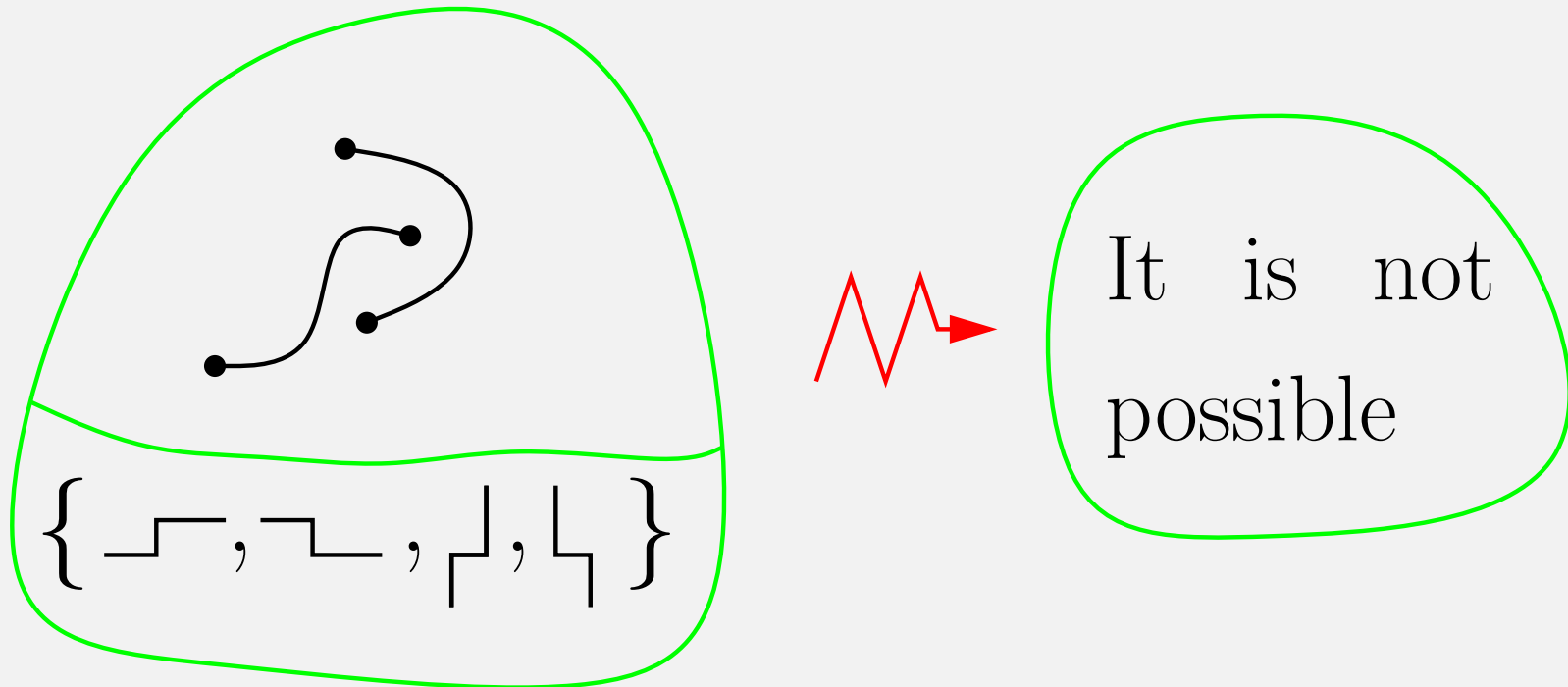


Map \equiv Collection of paths.

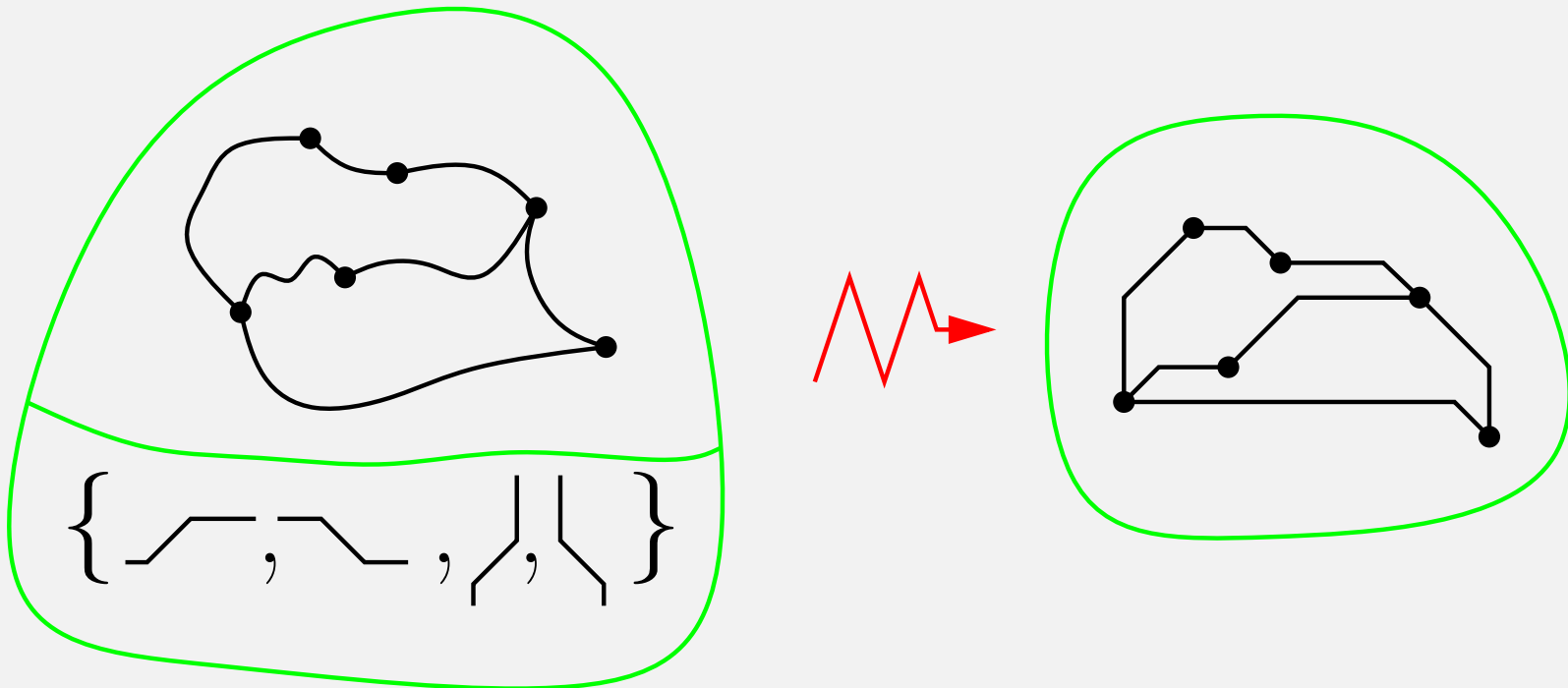
Our input and our output



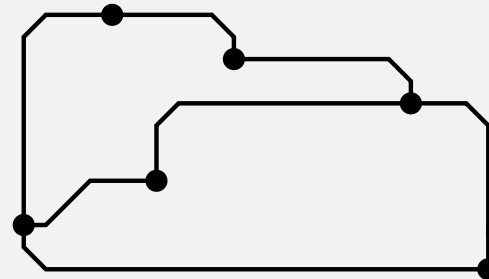
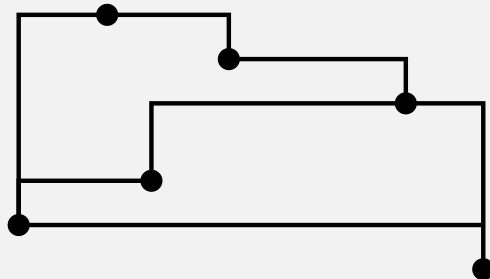
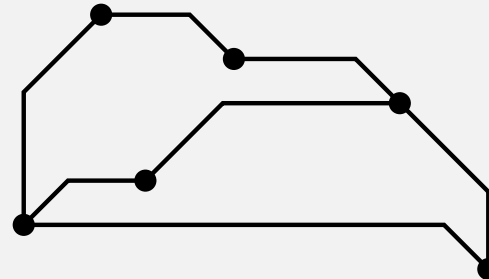
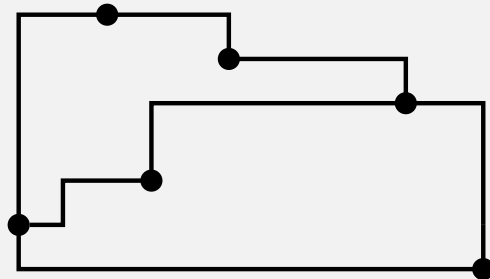
Our input and our output



Our input and our output



How nice are the results?

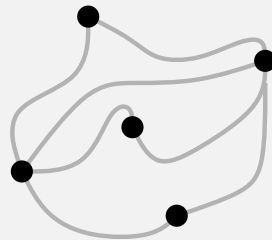
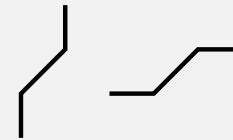


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:

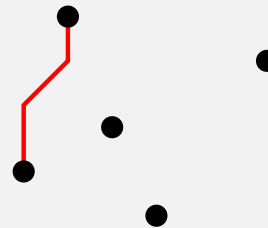
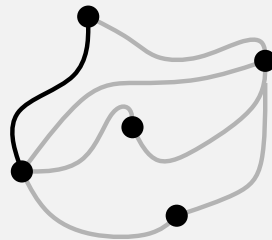
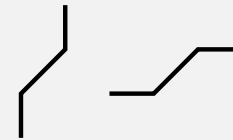


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:

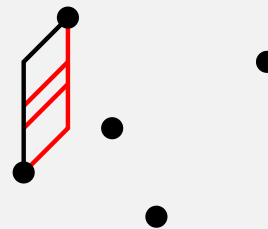
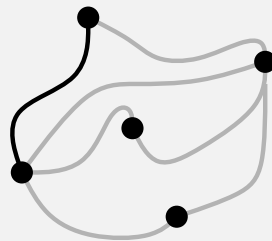
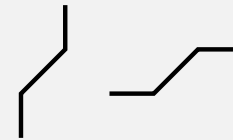


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:

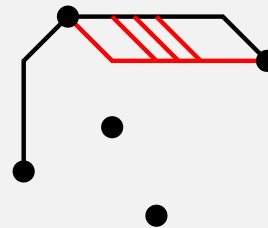
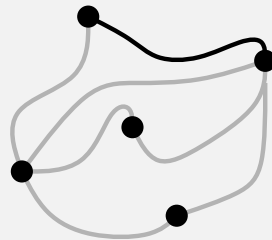
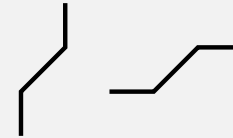


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:

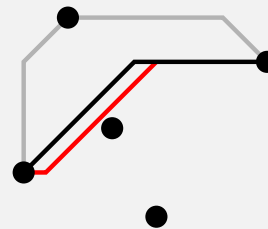
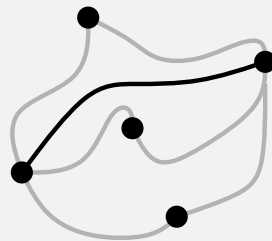
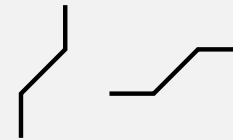


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:

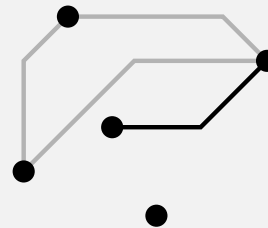
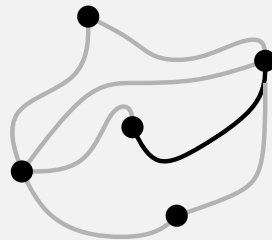
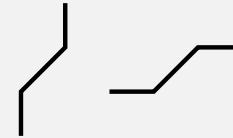


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:

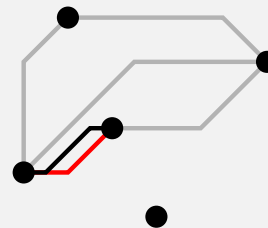
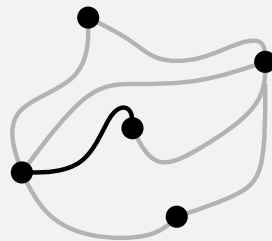
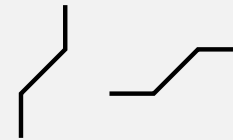


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:

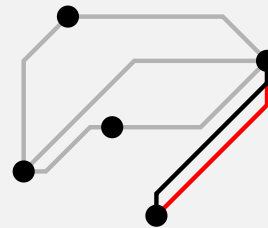
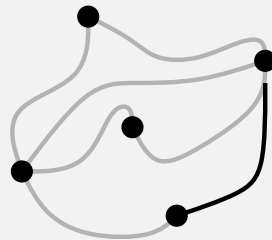
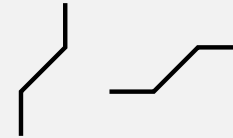


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:

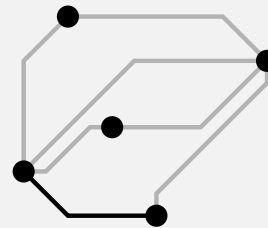
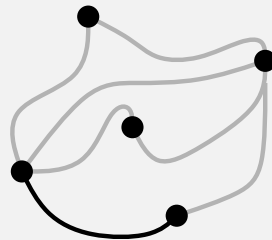
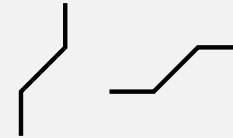


Idea behind everything

How to construct the schematic map?

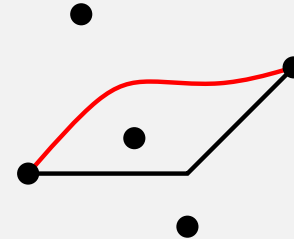
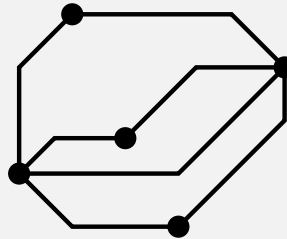
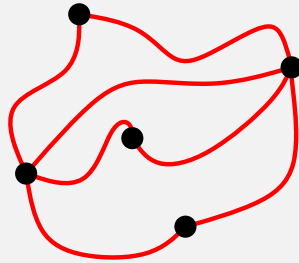
Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:



Schematic map: deformation of the original

We don't want this to happen

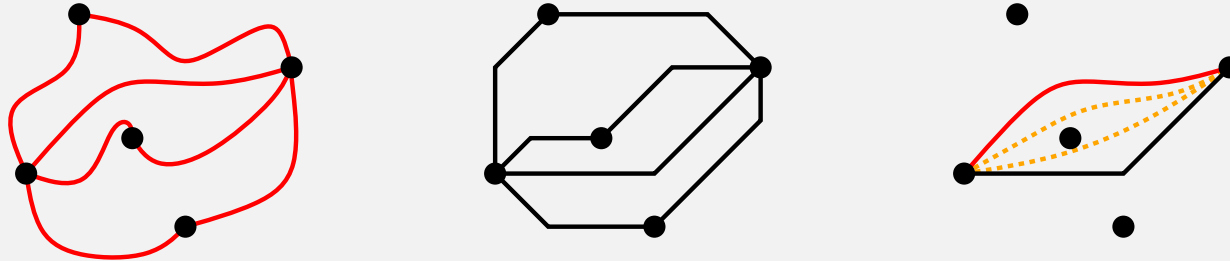


We want to keep cyclic order of paths around an endpoint.



Schematic map: deformation of the original

We don't want this to happen



We want to keep cyclic order of paths around an endpoint.

Each path of the schematic map is a deformation of the original path, without passing over the endpoints.

Or, each path in the original map is a deformation of a path in the schematic map, without passing over the endpoints.

Equivalent maps

Definition: Two maps M and M' are equivalent iff:

- they have the same endpoints,
- each path of M can be continuously deformed to a path of M' without passing over the endpoints.

Problem: Given M , find an equivalent M' with a given type of schematic connections.

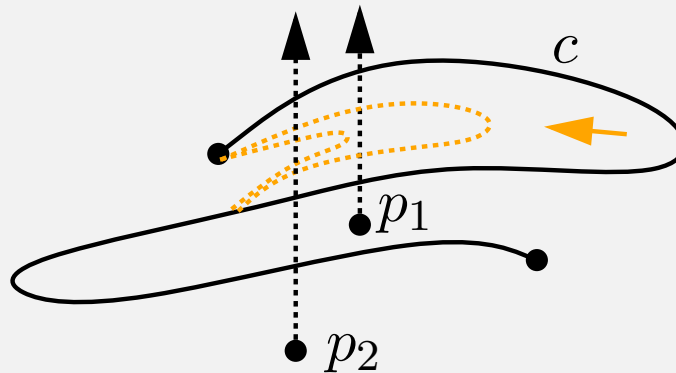
Basic question: Does our idea of top-to-bottom placement construct equivalent, schematic maps?

It depends on what means above and below.



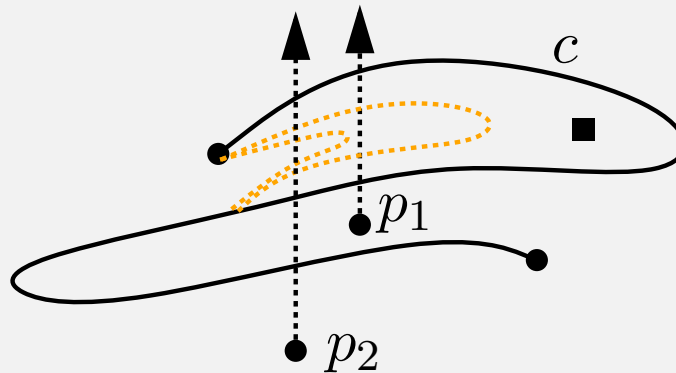
Points above and below a curve

Definition: Point p is below connection c if any continuous deformation of c that does not pass over p intersects the vertical upper ray from p .
Below is defined similarly.



Points above and below a curve

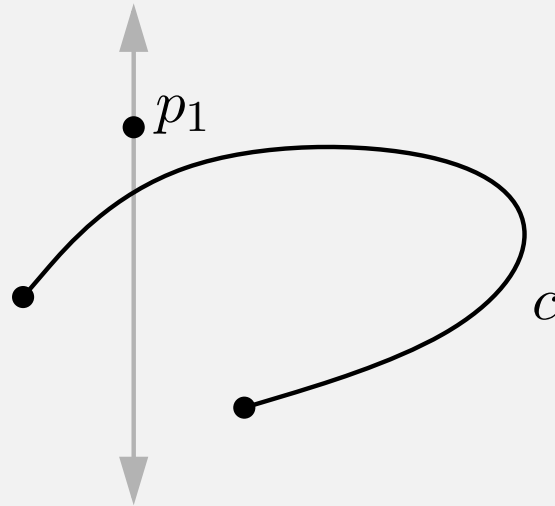
Definition: Point p is below connection c if any continuous deformation of c that does not pass over p intersects the vertical upper ray from p .
Below is defined similarly.



Remark: For comparing a point with a connection we don't worry of other points.

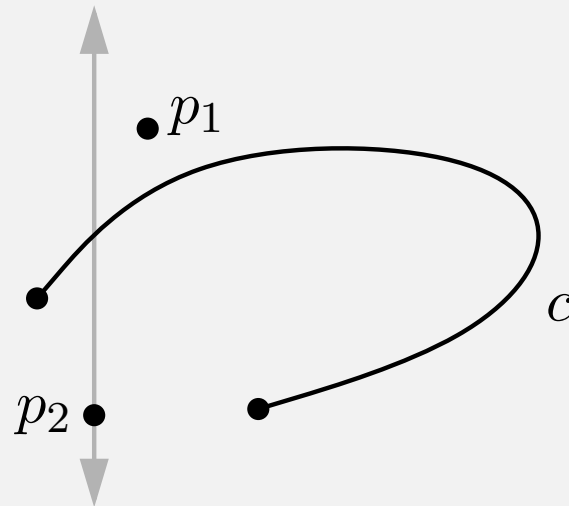


Example relations point-connection



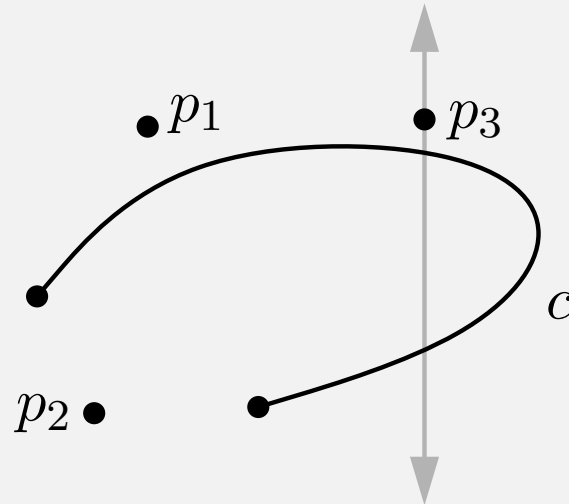
- p_1 is above c .

Example relations point-connection



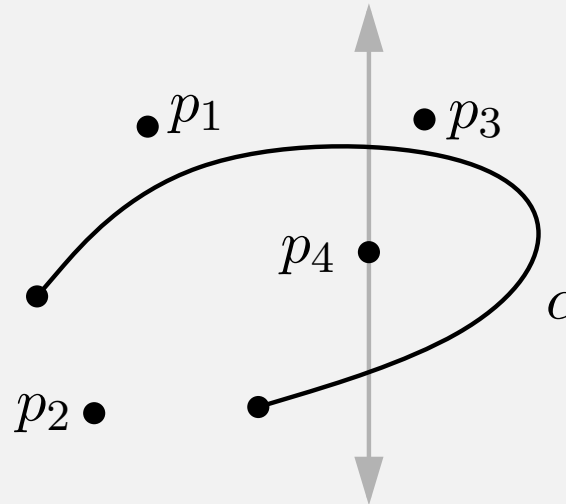
- p_1 is above c .
- p_2 is below c .

Example relations point-connection



- p_1 is above c .
- p_2 is below c .
- p_3 has no relation with c .

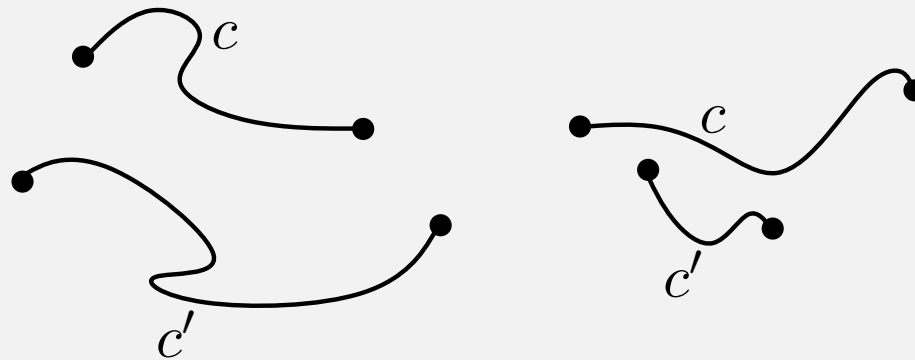
Example relations point-connection



- p_1 is above c .
- p_2 is below c .
- p_3 has no relation with c .
- p_4 is above and below c .

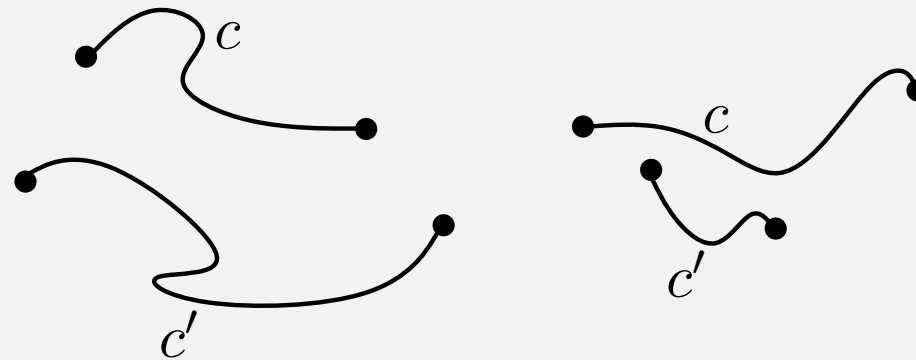
Relation connection-connection

Definition: Connection c is above connection c' if any endpoint of c is above c' or any endpoint of c' is below c



Relation connection-connection

Definition: Connection c is above connection c' if any endpoint of c is above c' or any endpoint of c' is below c



Property: Equivalent maps have the same order.

The order of the input map is preserved in the schematic one.

This is the top-to-bottom order we follow to place connections, leaving always as much freedom as possible.



Our algorithm

Input: Original map. Allowed types of connections.

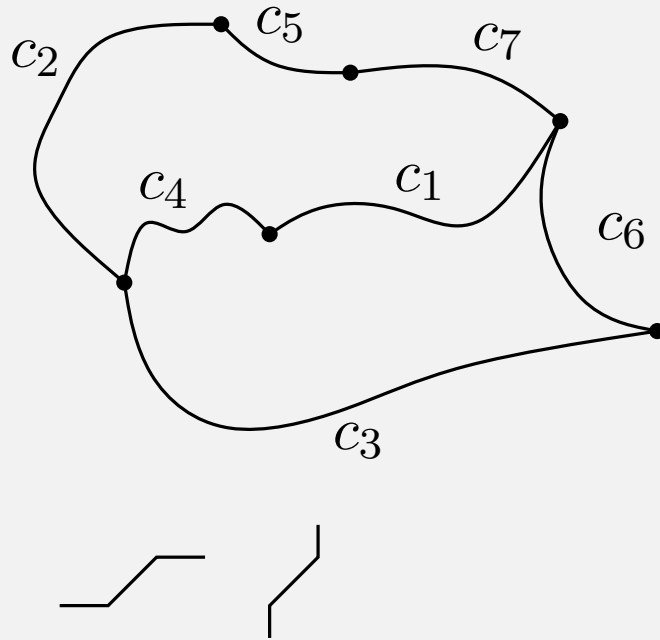
Output: Schematic map, or 'It is not possible'.

Procedure:

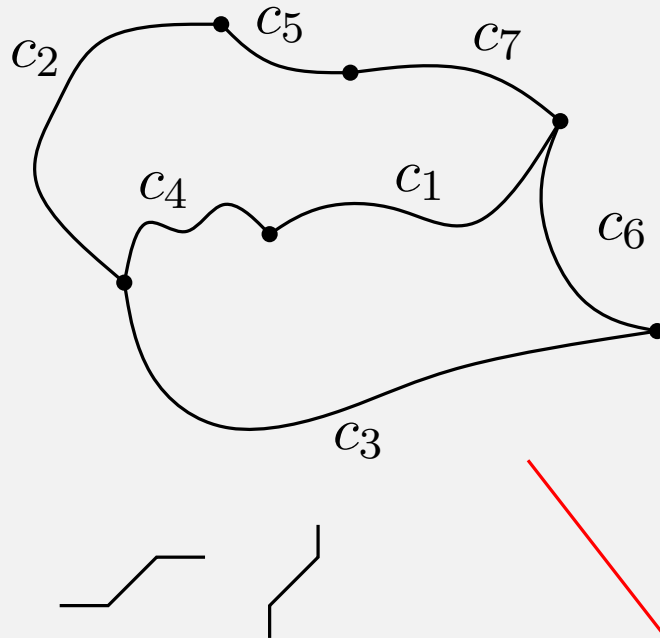
1. compute order among connections,
2. place connections (topmost each) in the computed order.



Example



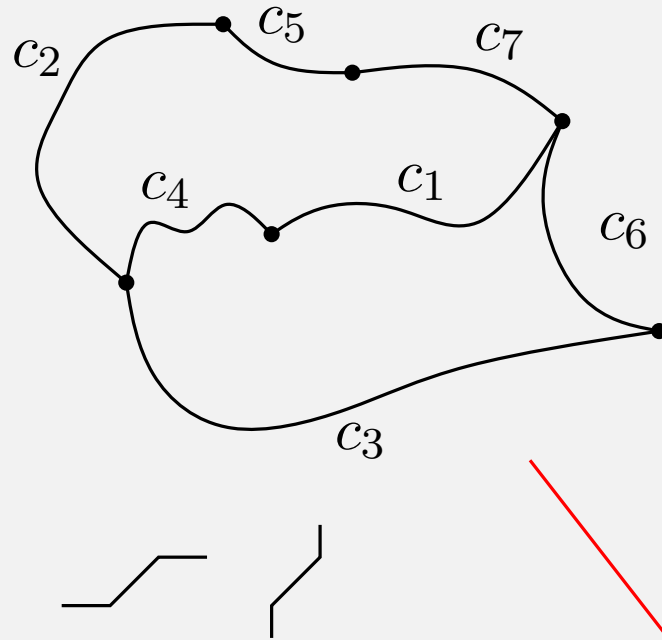
Example



$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$



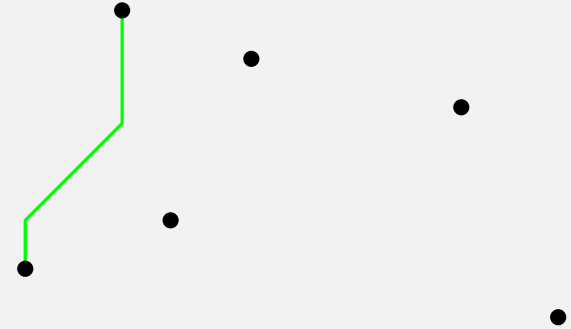
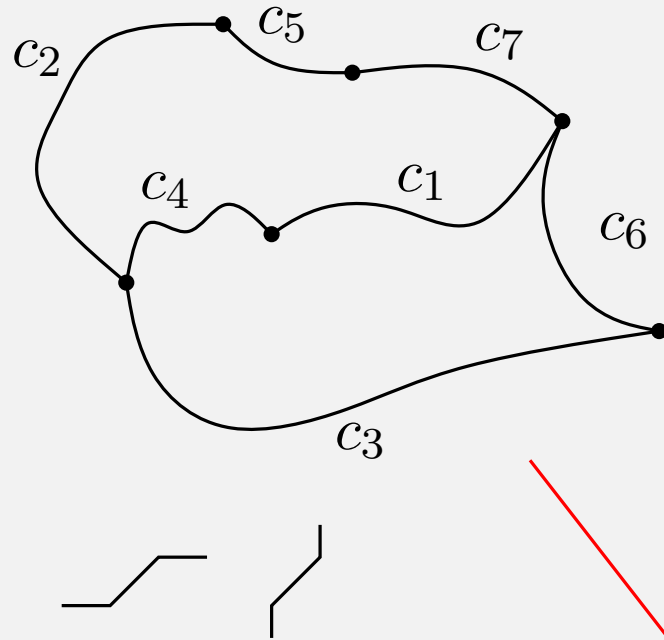
Example



$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$

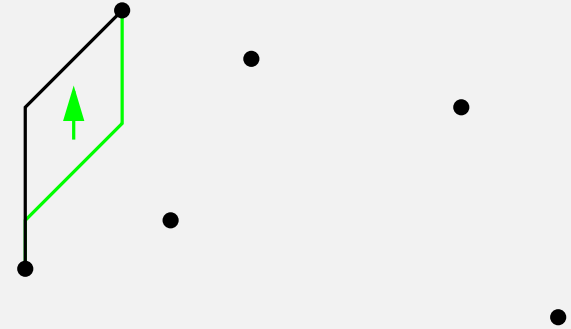
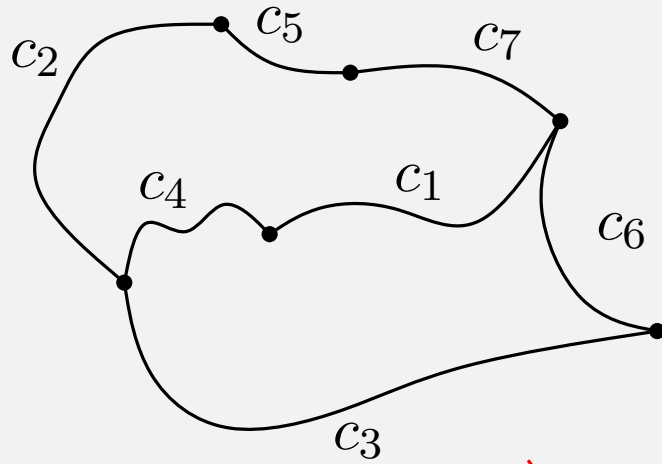


Example



$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$

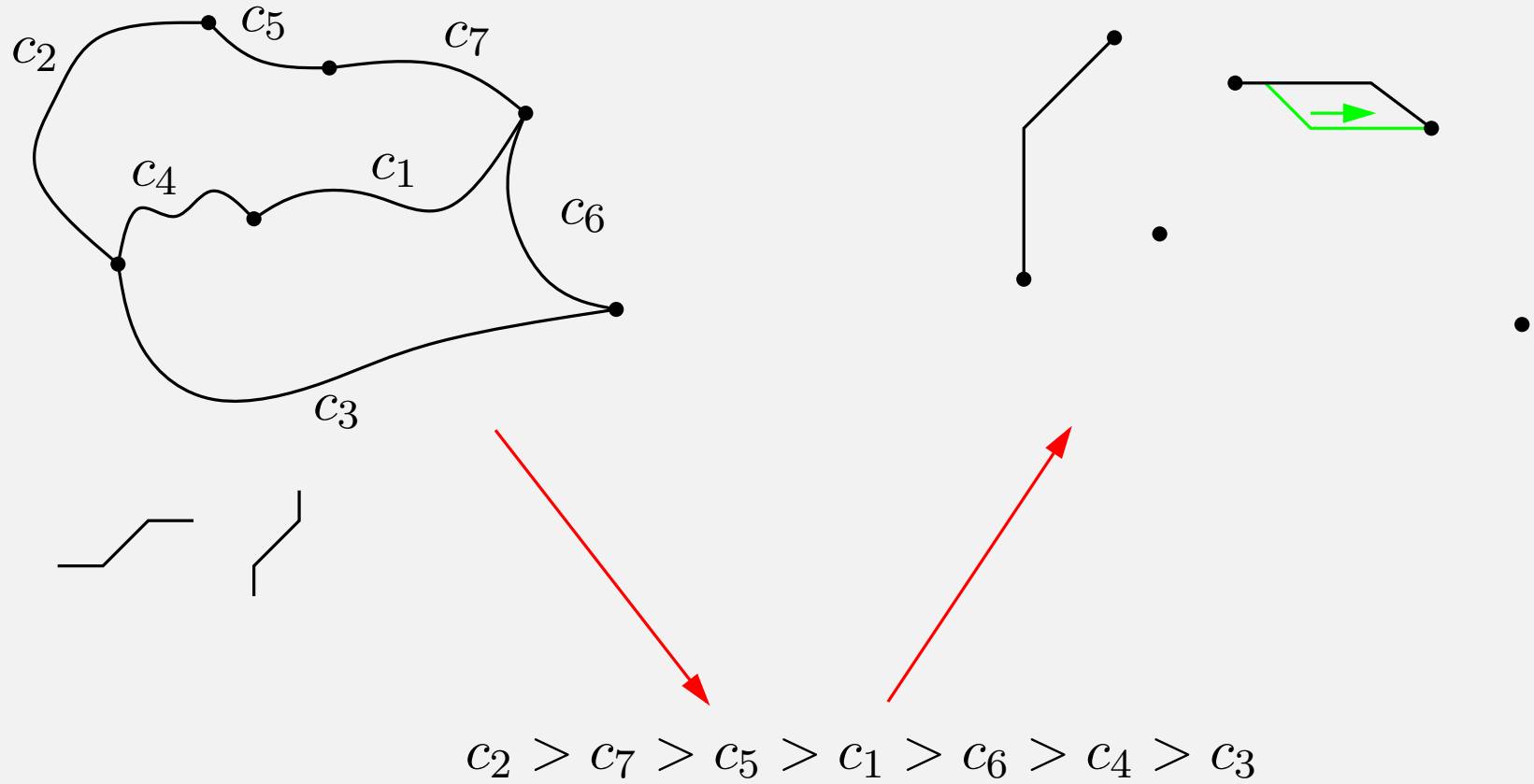
Example



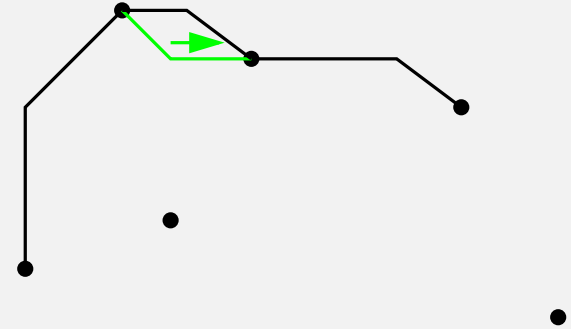
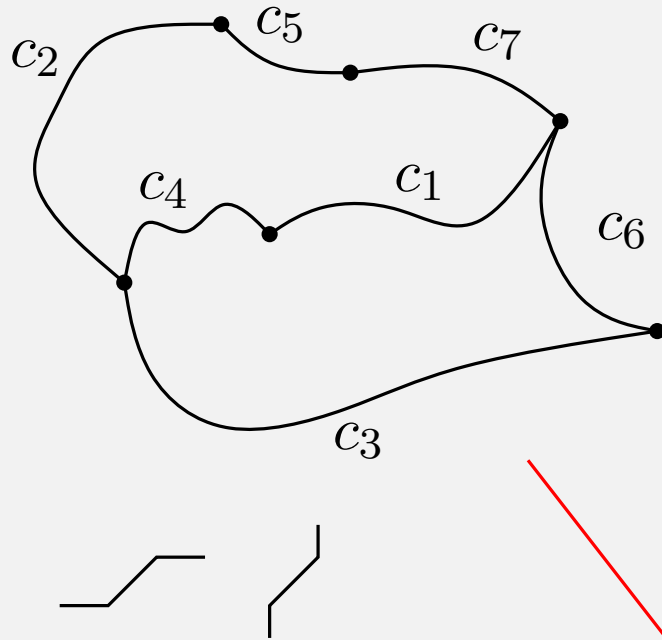
$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$



Example



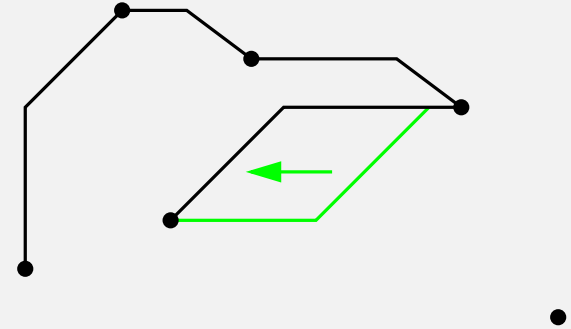
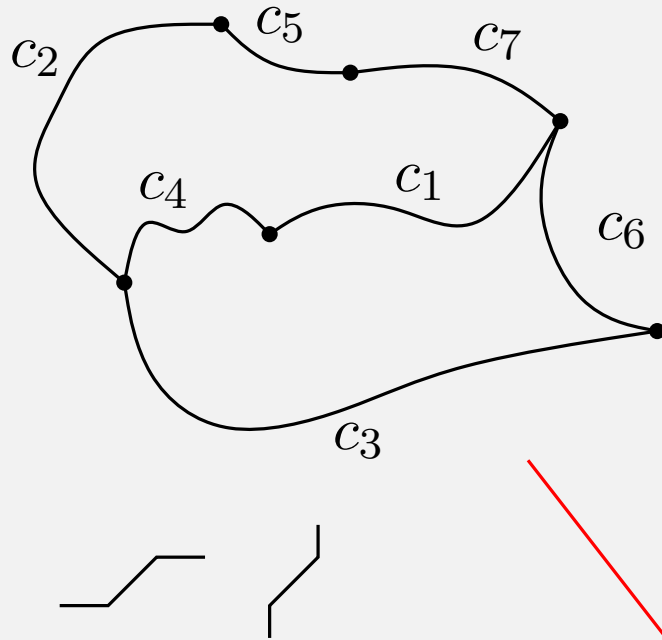
Example



$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$

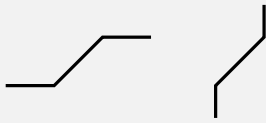
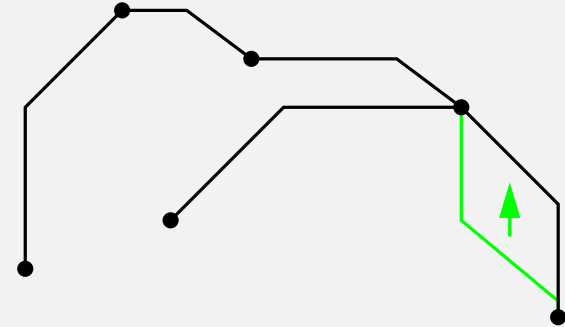
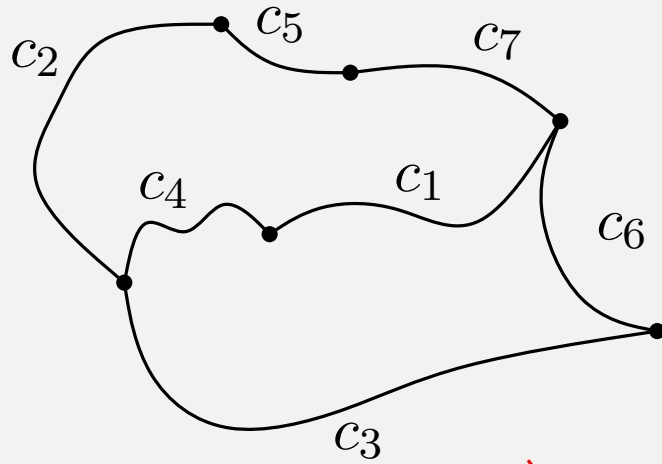


Example



$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$

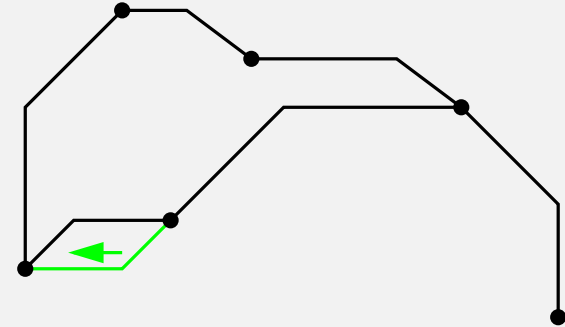
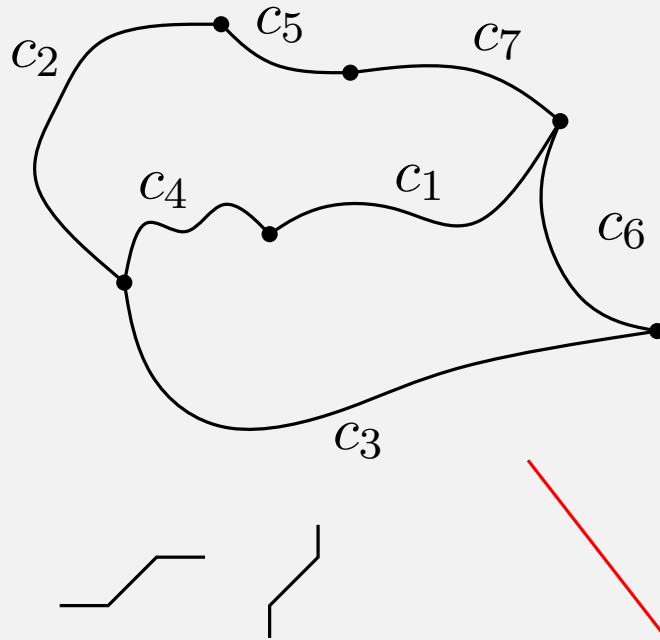
Example



$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$



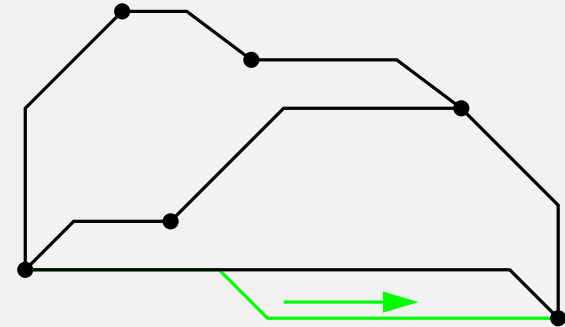
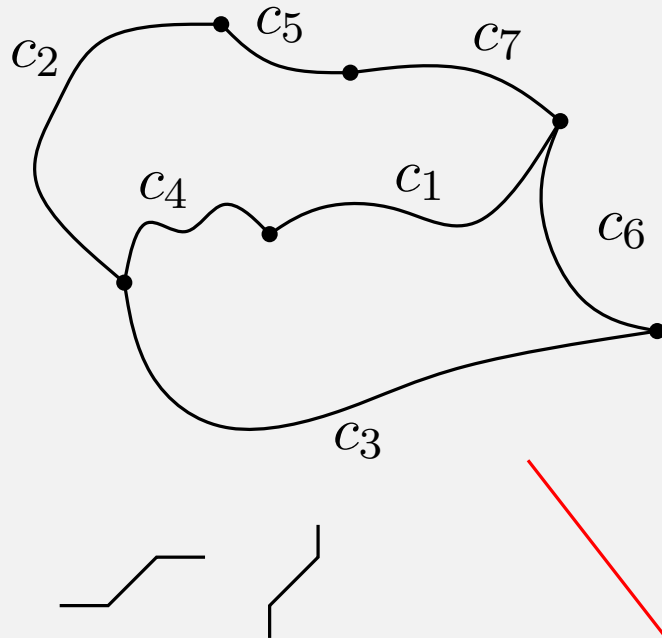
Example



$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$



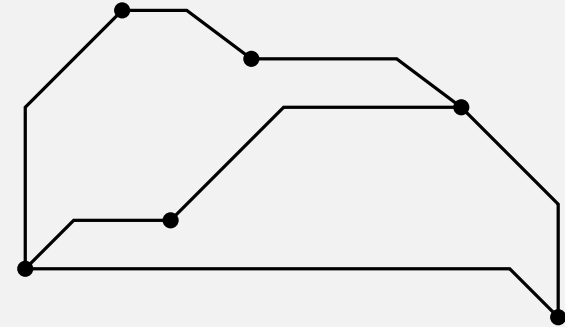
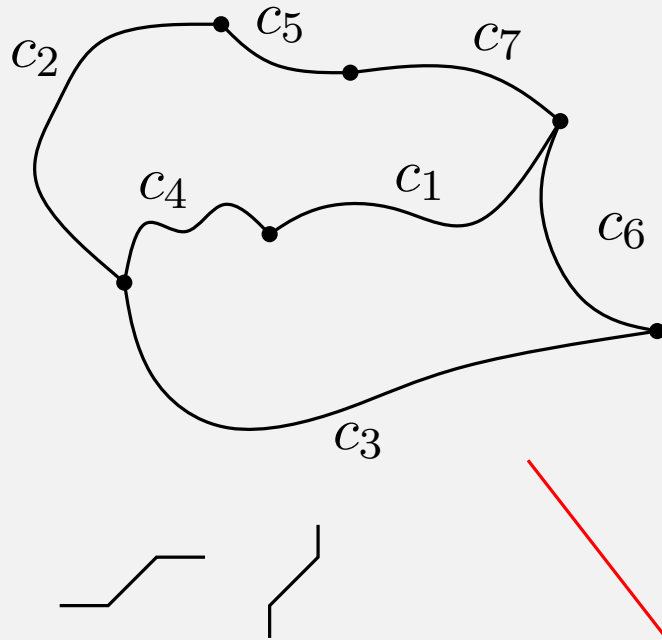
Example



$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$



Example

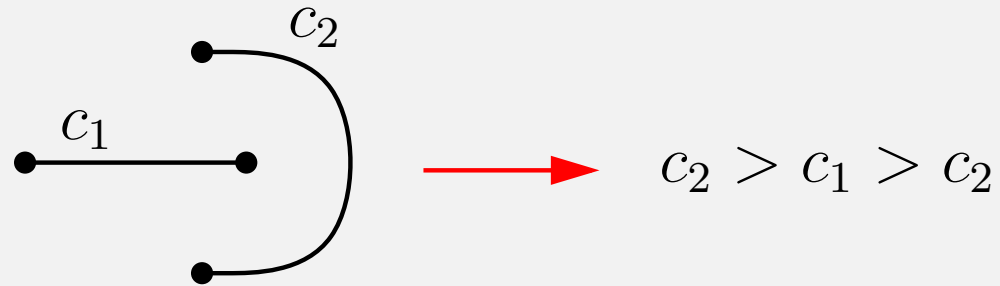


$$c_2 > c_7 > c_5 > c_1 > c_6 > c_4 > c_3$$



Possible problems

- We have a cyclic order.

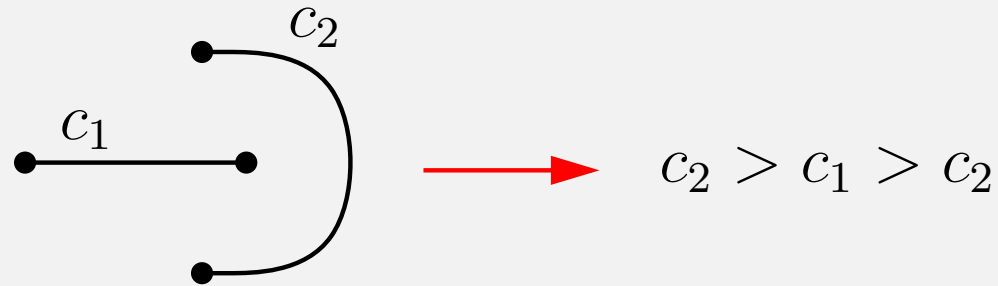


Which one do we place first?



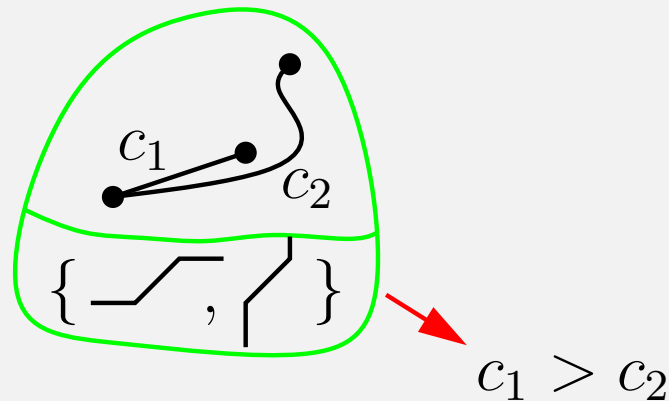
Possible problems

- We have a cyclic order.



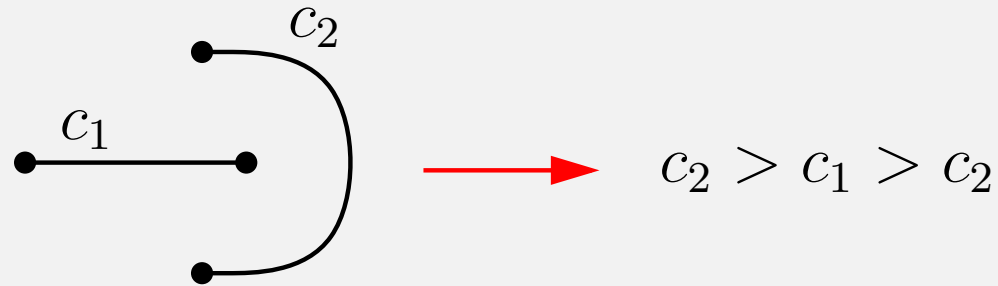
Which one do we place first?

- We cannot place next connection.



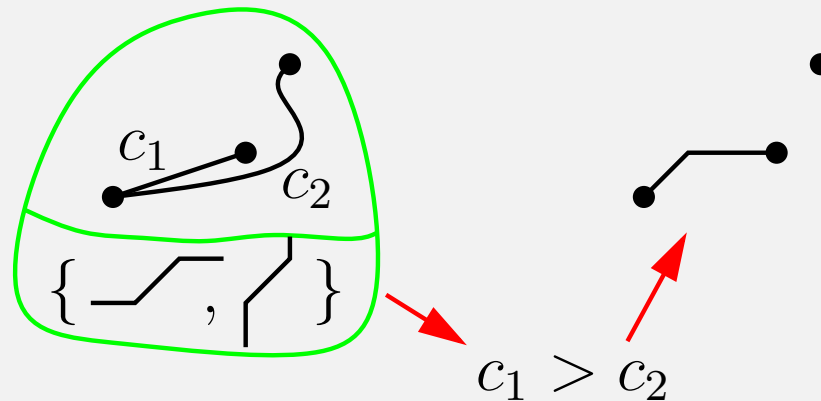
Possible problems

- We have a cyclic order.



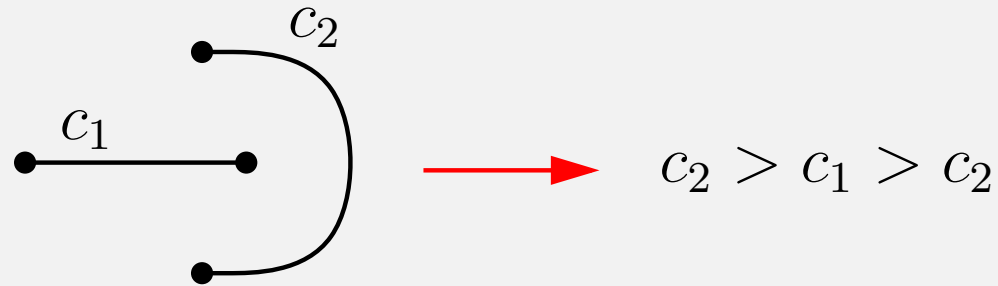
Which one do we place first?

- We cannot place next connection.



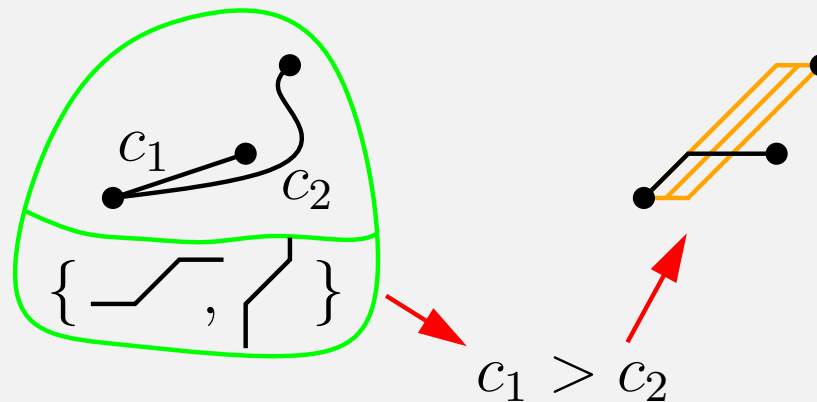
Possible problems

- We have a cyclic order.



Which one do we place first?

- We cannot place next connection.



Things left for now

- How to compute the order quickly?
- How to place each connection topmost?
- Which combinations of connections can we handle?



How to compute the order quickly?

Naive approach: Compute the relation for each pair of connections.



How to compute the order quickly?

Naive approach: Compute the relation for each pair of connections.

More advanced approach: Use a data structure (oracle) that querying with a connection gives you another connection above.



How to compute the order quickly?

Naive approach: Compute the relation for each pair of connections.

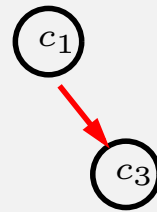
More advanced approach: Use a data structure (oracle) that querying with a connection gives you another connection above.



How to compute the order quickly?

Naive approach: Compute the relation for each pair of connections.

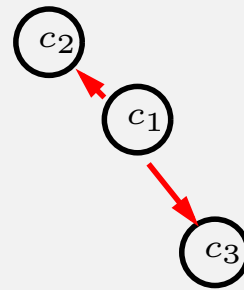
More advanced approach: Use a data structure (oracle) that querying with a connection gives you another connection above.



How to compute the order quickly?

Naive approach: Compute the relation for each pair of connections.

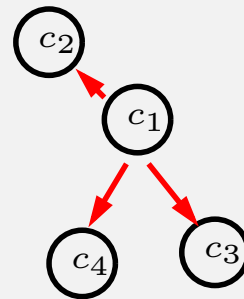
More advanced approach: Use a data structure (oracle) that querying with a connection gives you another connection above.



How to compute the order quickly?

Naive approach: Compute the relation for each pair of connections.

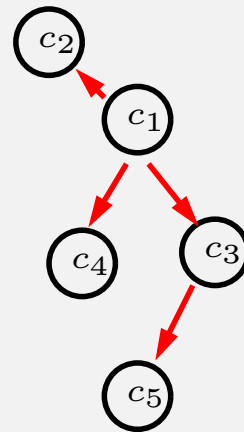
More advanced approach: Use a data structure (oracle) that querying with a connection gives you another connection above.



How to compute the order quickly?

Naive approach: Compute the relation for each pair of connections.

More advanced approach: Use a data structure (oracle) that querying with a connection gives you another connection above.



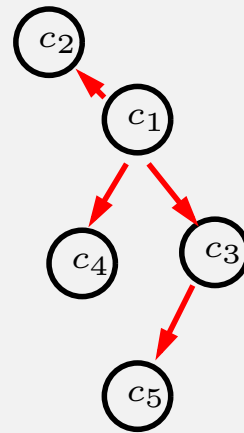
Order recursively connections above and below c_1 .



How to compute the order quickly?

Naive approach: Compute the relation for each pair of connections.

More advanced approach: Use a data structure (oracle) that querying with a connection gives you another connection above.



Order recursively connections above and below c_1 .

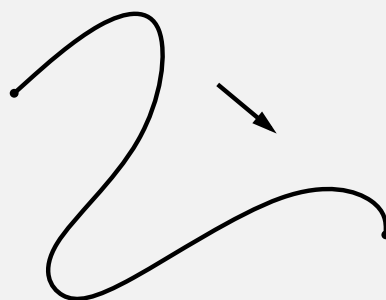
Some restricted maps: Better ways...



How to compute the order quickly?

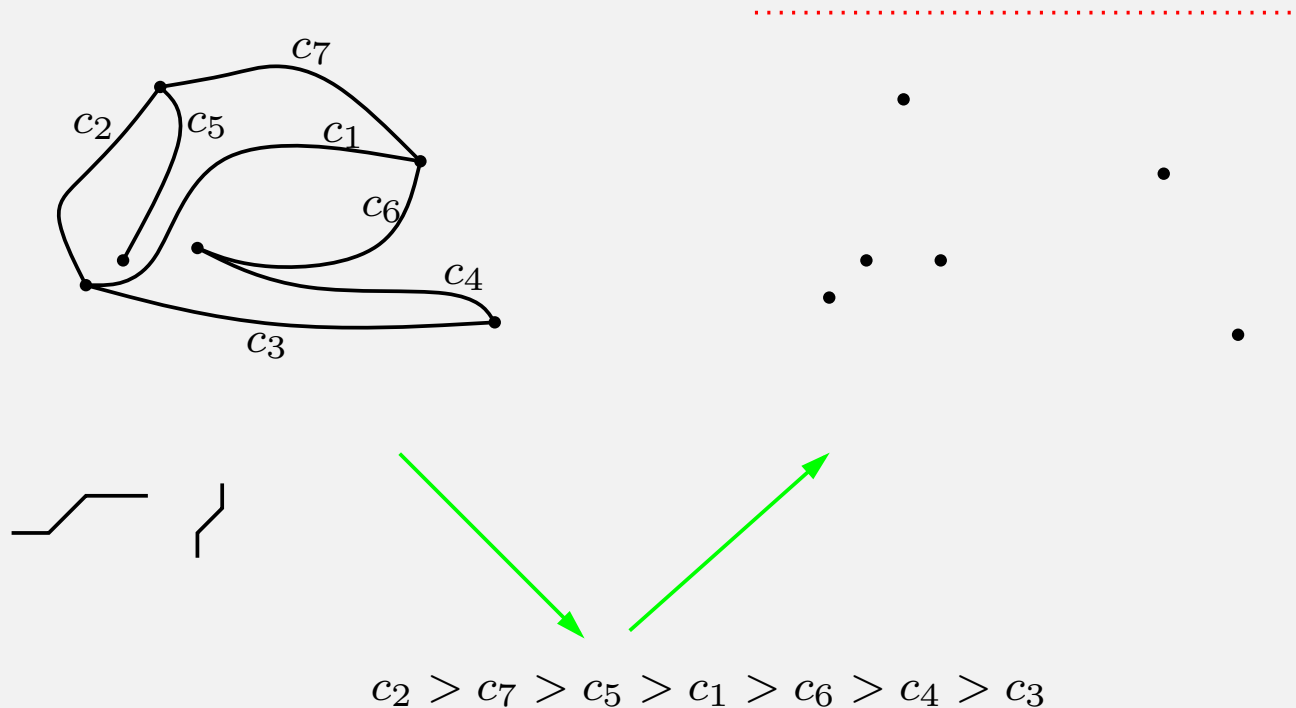
If the input map has n segments:

- **Naive approach:** $O(n^2)$ time.
The worst case is proportional to n^2 .
- **More advanced approach:** $O(n \log^3 n)$ time.
The worst case is proportional to $n \log^3 n$.
- **Map with monotone paths:** $O(n \log n)$ time.
The worst case is proportional to $n \log n$.



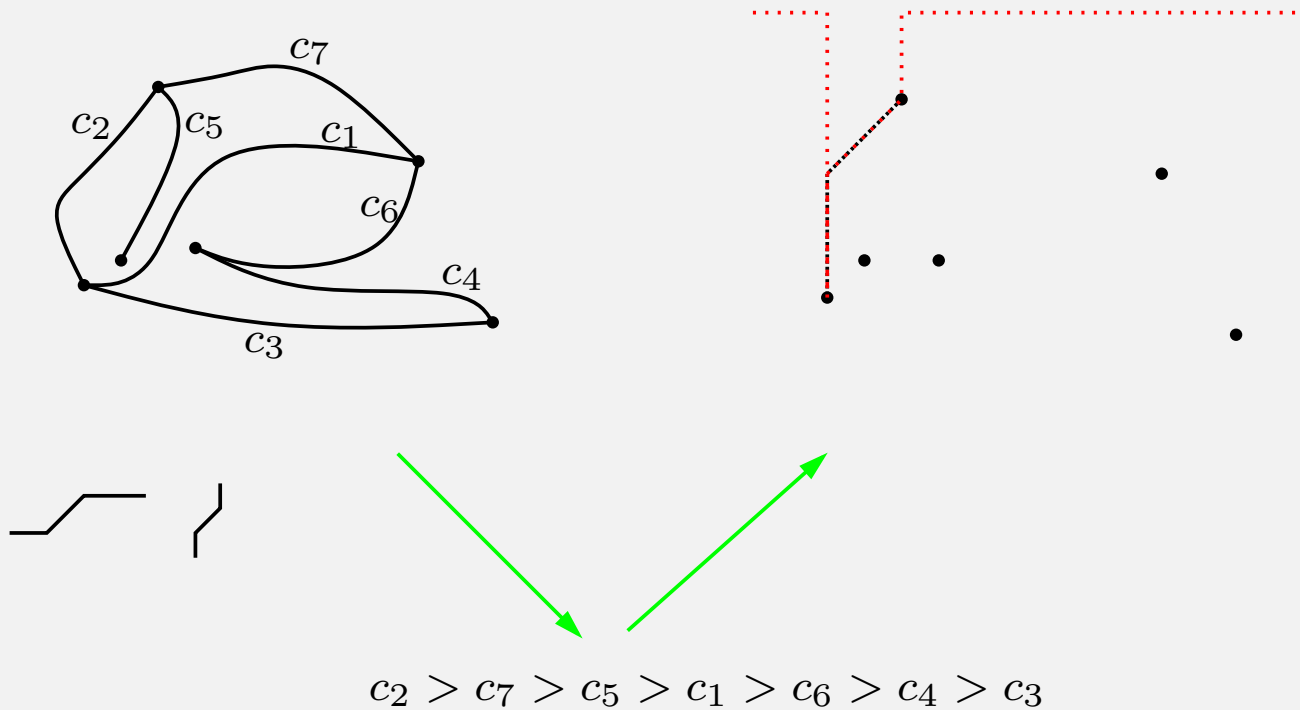
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



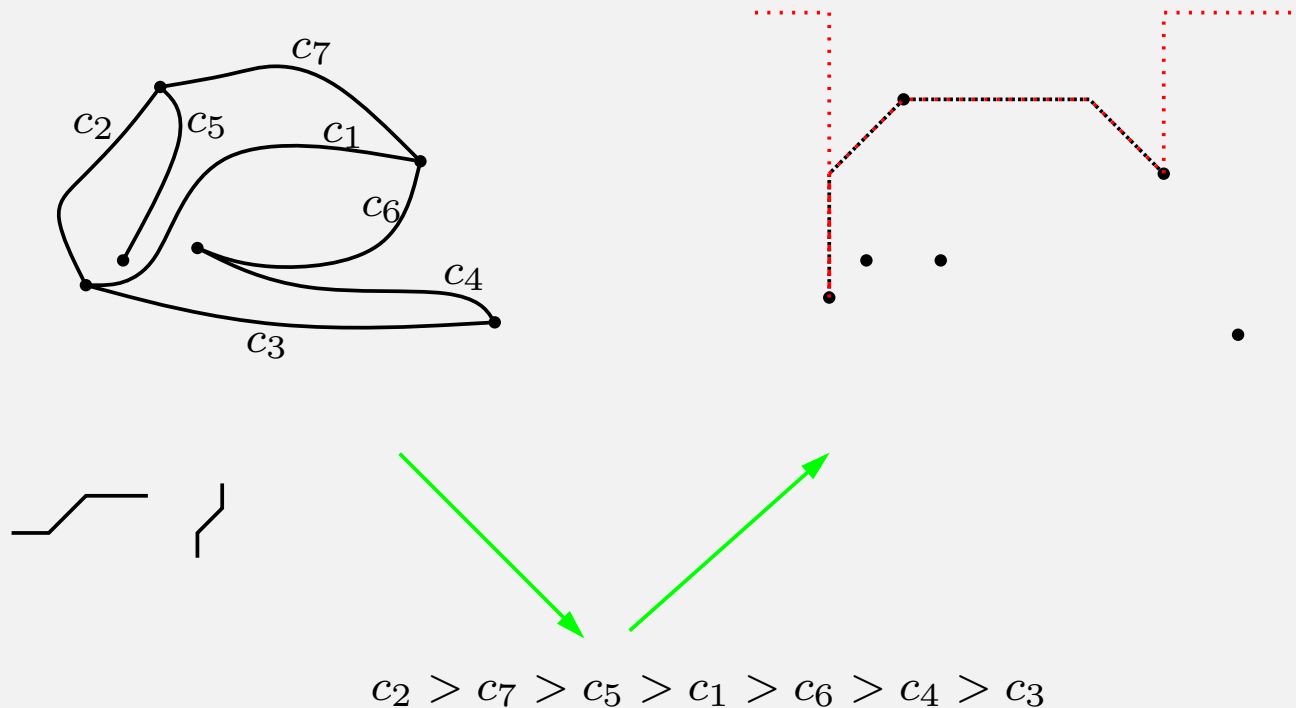
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



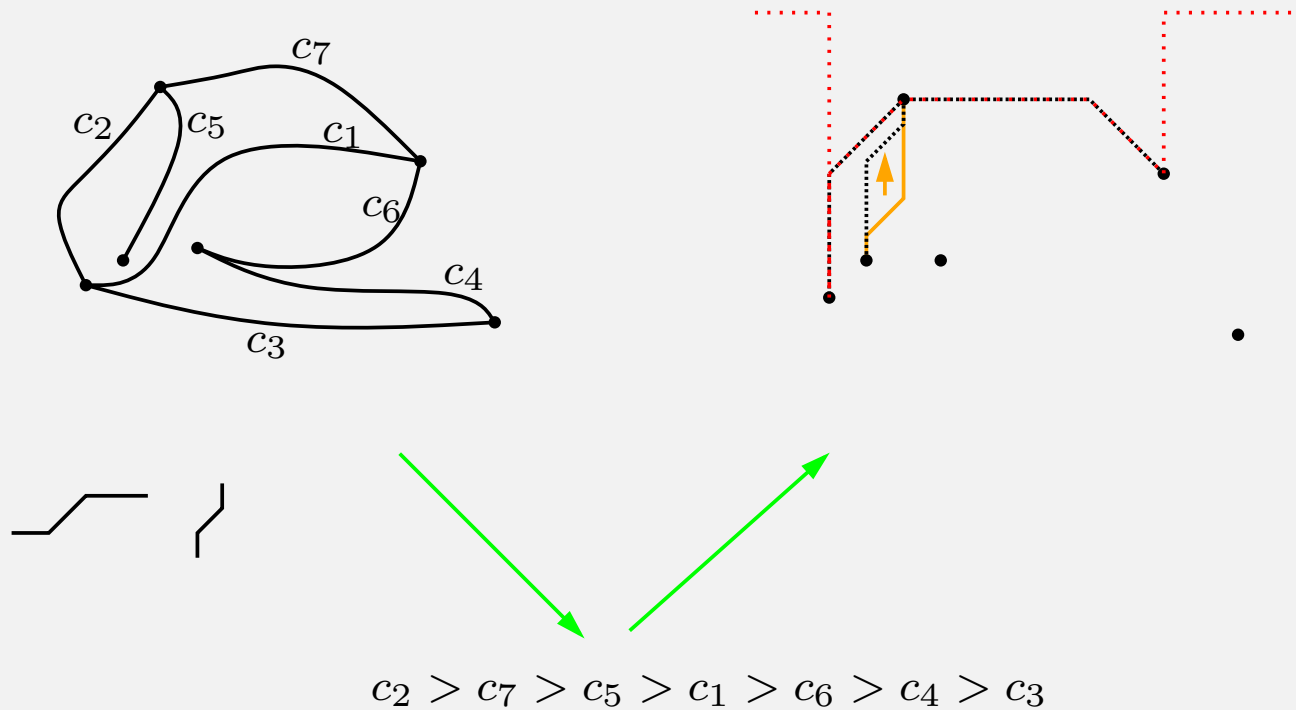
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



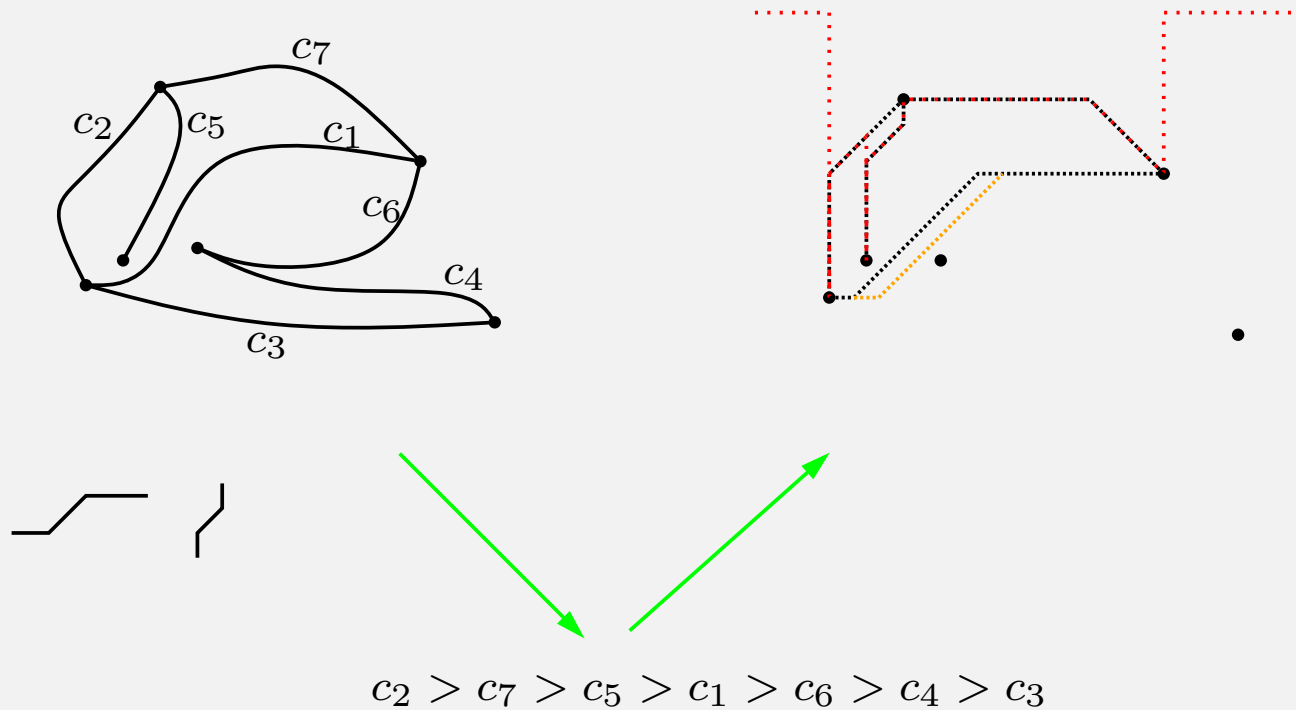
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



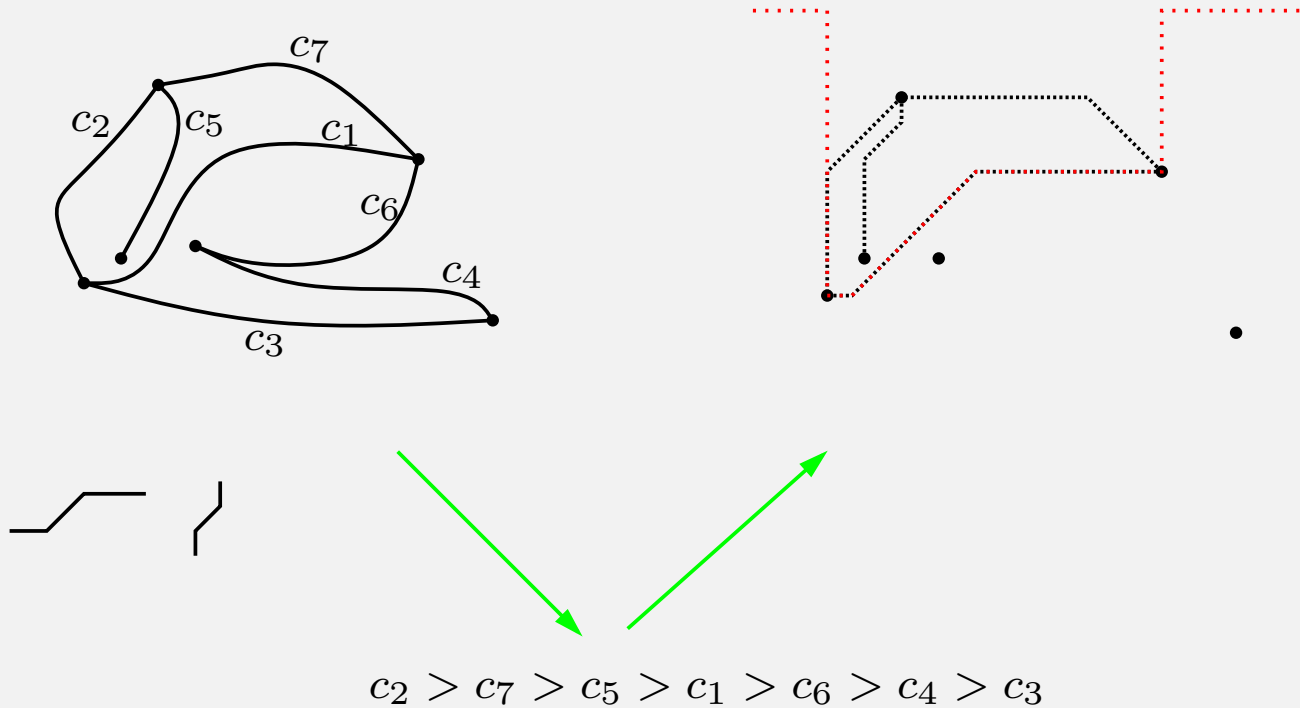
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



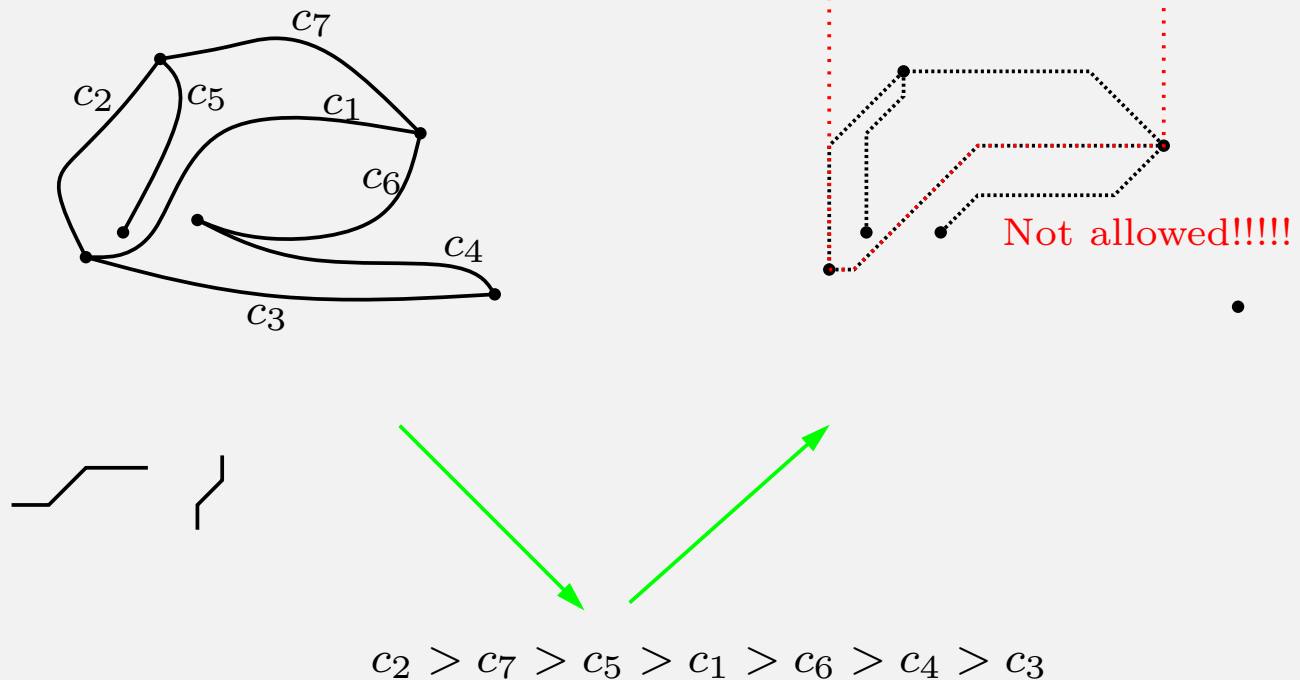
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



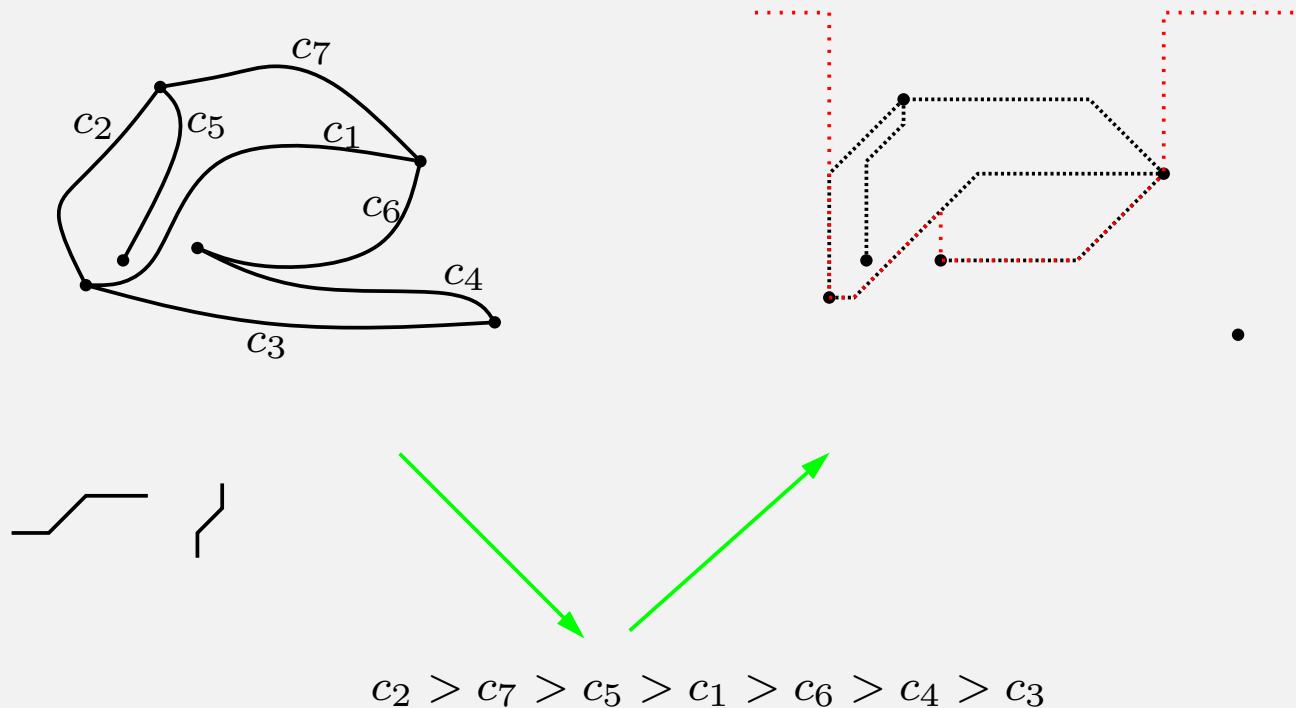
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



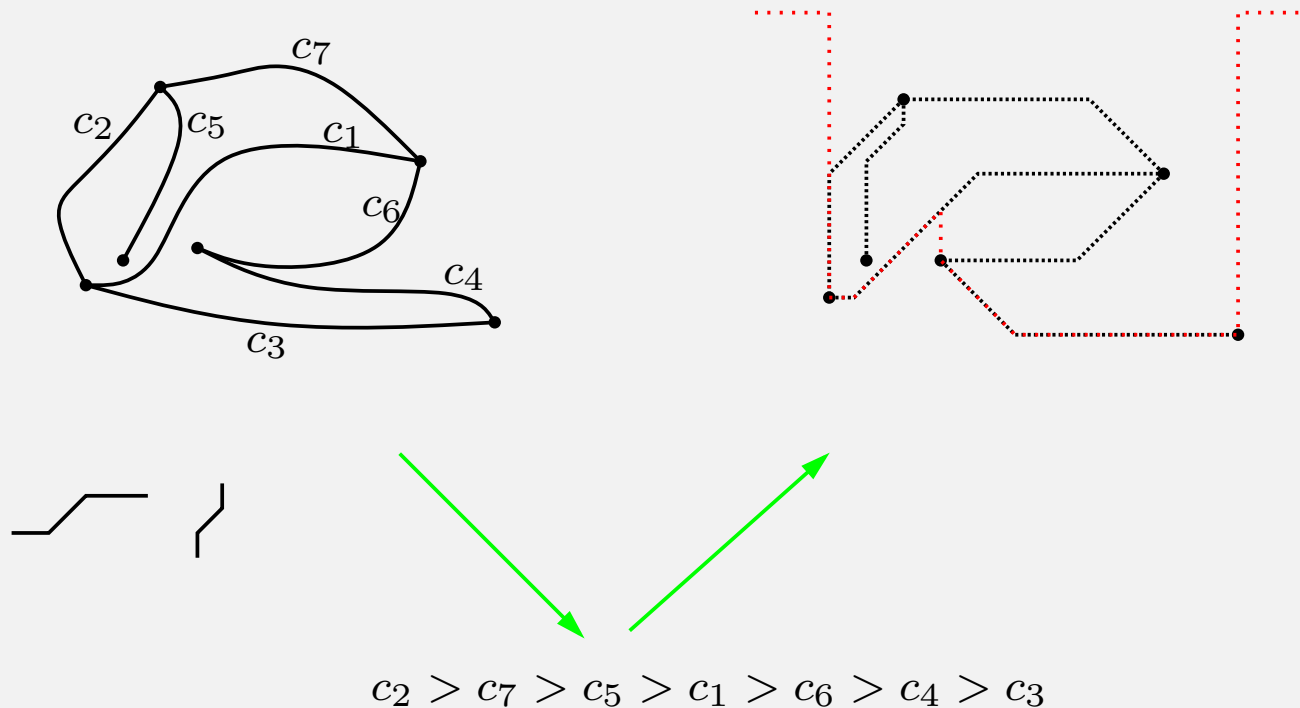
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



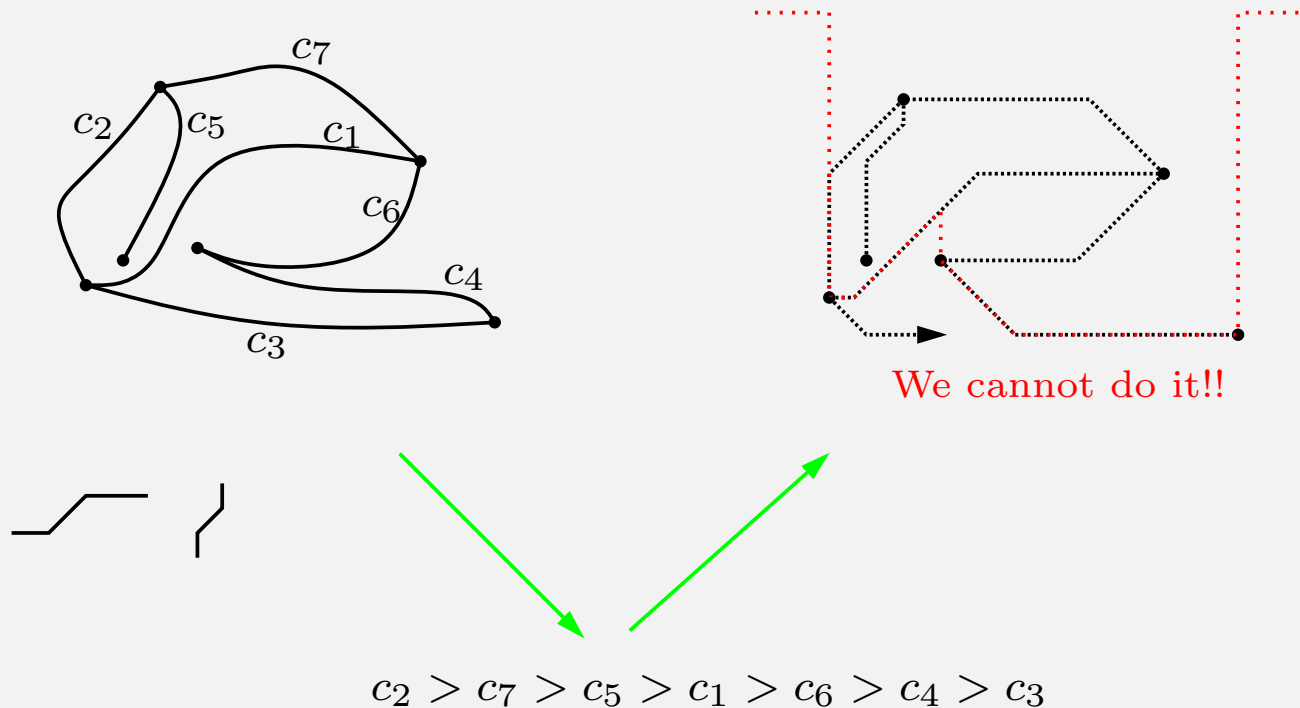
How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.



How to place each connection topmost?

When we are placing each connection we maintain the lower envelope of the already-placed connections.

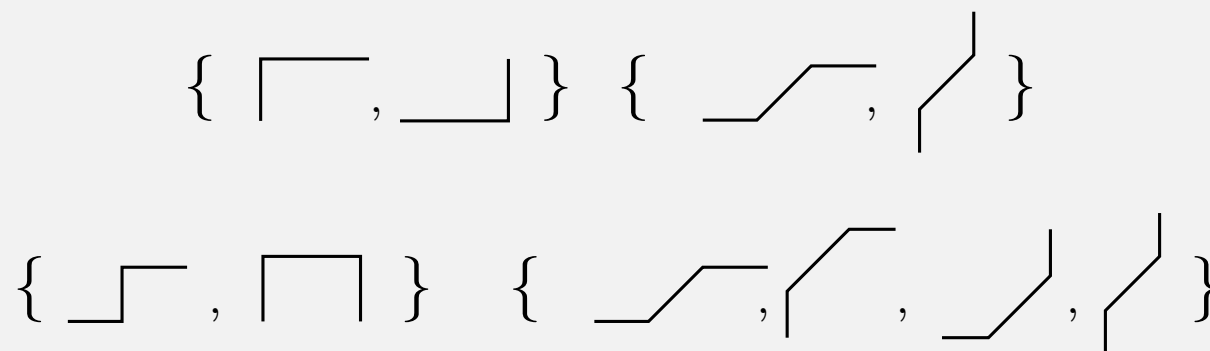


Which combinations of connections can we handle?

We only can use types of connections that have a clear topmost placement for any lower envelope.



Examples of this



Work to do

- Compute order more quickly.
- Handle connections with topmost not defined.
- Experimental results.

