Local certification of planarity

Laurent Feuilloley

based on
Compact Distributed Certification of Planar Graphs
joint work with Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Éric Rémila and Ioan Todinca.

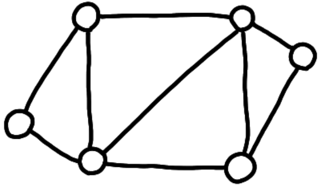
GRAA Seminar · 18th June 2020

Distributed decision

Problem : Is the graph in the class X ?

1. Max degree ≤ 5
2. Paths
3. Planar

Model : distributed decision.

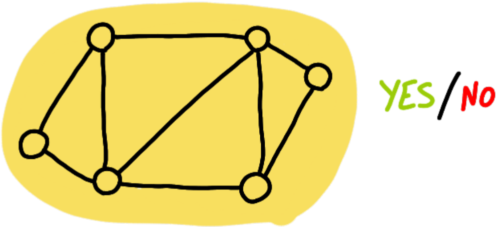


Distributed decision

Problem : Is the graph in the class X ?

1. Max degree ≤ 5
2. Paths
3. Planar

Model : distributed decision.

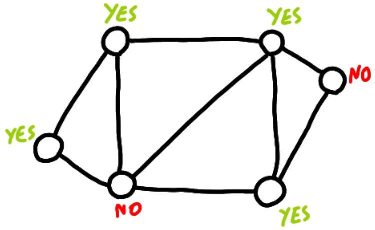


Distributed decision

Problem : Is the graph in the class X ?

1. Max degree ≤ 5
2. Paths
3. Planar

Model : distributed decision.



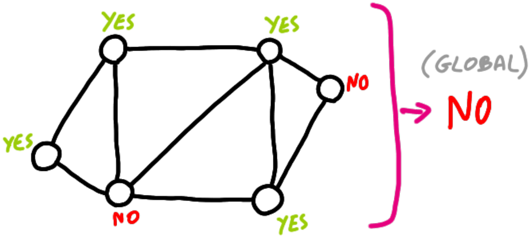
Decision rule : Global YES $\Leftrightarrow$ All nodes say YES

Distributed decision

Problem : Is the graph in the class X ?

1. Max degree ≤ 5
2. Paths
3. Planar

Model : distributed decision.



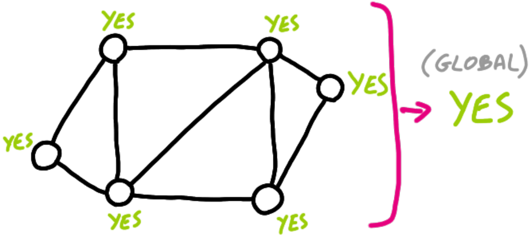
Decision rule : Global YES $\Leftrightarrow$ All nodes say YES

Distributed decision

Problem : Is the graph in the class X ?

1. Max degree ≤ 5
2. Paths
3. Planar

Model : distributed decision.



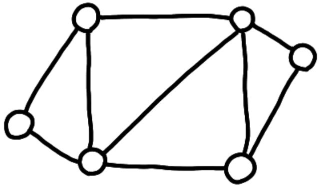
Decision rule : Global YES $\Leftrightarrow$ All nodes say YES

Basic local decision

The basic mechanism :

1. all nodes wake up at the same time
2. look at their neighbors
3. run an algorithm to choose an output

[Note : complexity of the algorithm not considered.]

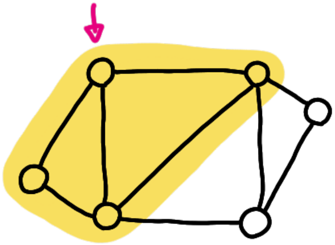


Basic local decision

The basic mechanism :

1. all nodes wake up at the same time
2. look at their neighbors
3. run an algorithm to choose an output

[Note : complexity of the algorithm not considered.]

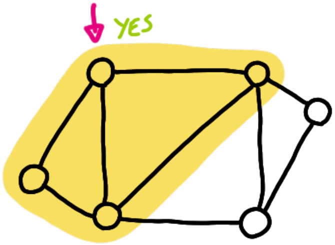


Basic local decision

The basic mechanism :

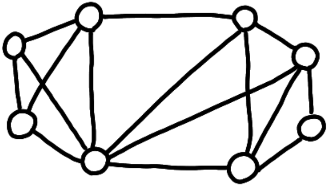
1. all nodes wake up at the same time
2. look at their neighbors
3. run an algorithm to choose an output

[Note : complexity of the algorithm not considered.]



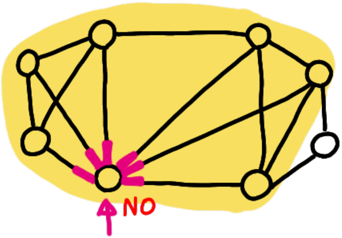
Example 1

Problem : Is the graph in the class "degree at most 5" ?
Algorithm (run at all nodes) :
If my degree is 5 or less : YES.
Othewise NO.



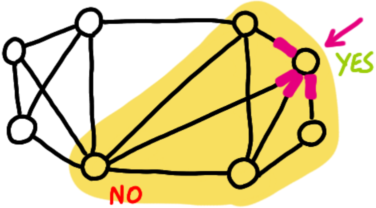
Example 1

Problem : Is the graph in the class "degree at most 5" ?
Algorithm (run at all nodes) :
If my degree is 5 or less : YES.
Othewise NO.



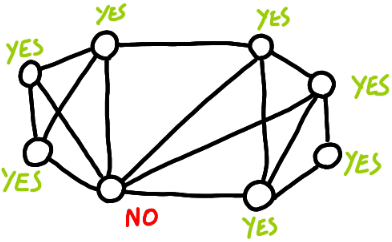
Example 1

Problem : Is the graph in the class "degree at most 5" ?
Algorithm (run at all nodes) :
If my degree is 5 or less : YES.
Othewise NO.



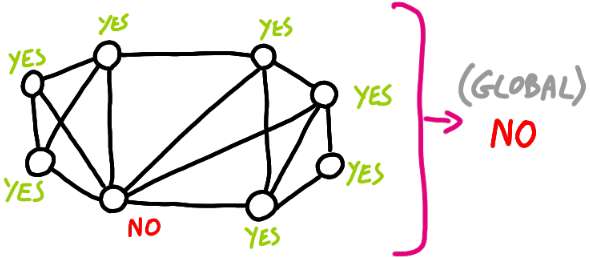
Example 1

Problem : Is the graph in the class "degree at most 5" ?
Algorithm (run at all nodes) :
If my degree is 5 or less : YES.
Othewise NO.



Example 1

Problem : Is the graph in the class "degree at most 5" ?
Algorithm (run at all nodes) :
If my degree is 5 or less : YES.
Othewise NO.

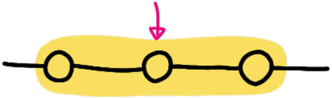


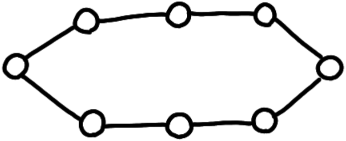
Limits of the basic decision

Problem : Is the graph a path ?

[Note : the graph is assumed to be connected.]

Theorem : No local algorithm can decide this class.

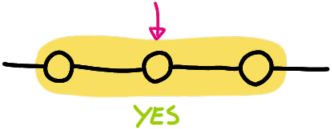


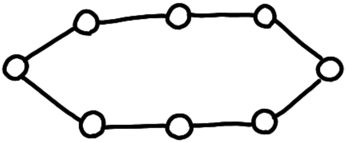


Limits of the basic decision

Problem : Is the graph a path ?
[Note : the graph is assumed to be connected.]

Theorem : No local algorithm can decide this class.

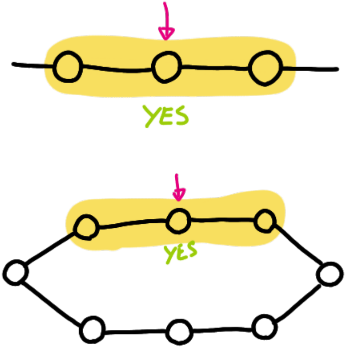




Limits of the basic decision

Problem : Is the graph a path ?
[Note : the graph is assumed to be connected.]

Theorem : No local algorithm can decide this class.

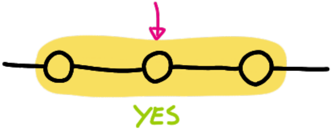


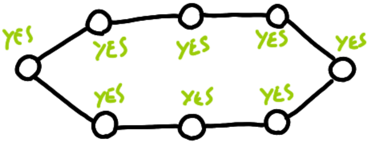
Limits of the basic decision

Problem : Is the graph a path ?

[Note : the graph is assumed to be connected.]

Theorem : No local algorithm can decide this class.

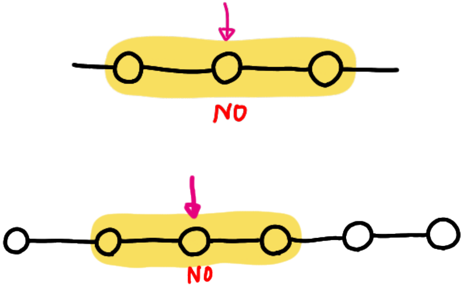




Limits of the basic decision

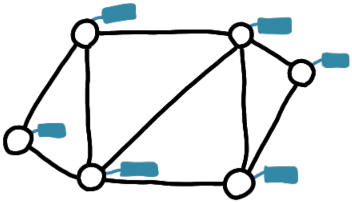
Problem : Is the graph a path ?
[Note : the graph is assumed to be connected.]

Theorem : No local algorithm can decide this class.



Local certification

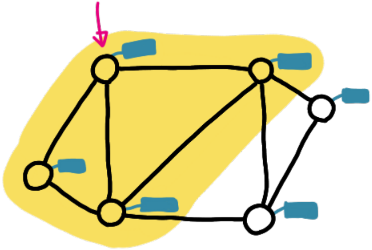
New thing : a labeling of the nodes ($\ell : V \rightarrow \{0,1\}^*$)



Definition : A scheme recognizes the class X if :
there exists a local algorithm such that $\forall G$:
$G \in X \Leftrightarrow$ there exists ℓ such that A accept G with labeling ℓ.

Local certification

New thing : a labeling of the nodes ($\ell : V \to \{0,1\}^*$)

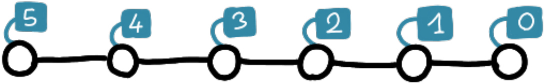


Definition : A scheme recognizes the class X if :
there exists a local algorithm such that $\forall G$:
$G \in X \Leftrightarrow$ there exists ℓ such that A accept G with labeling ℓ.

Example 2

Problem : Is the graph a path ?
Algorithm :

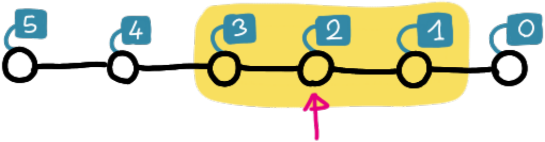
1. Check degree 1 or 2
2. Interpret labels as distances to a root and check consistency.



Example 2

Problem : Is the graph a path ?
Algorithm :

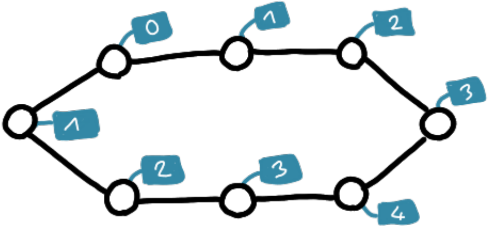
1. Check degree 1 or 2
2. Interpret labels as distances to a root and check consistency.



Example 2

Problem : Is the graph a path ?
Algorithm :

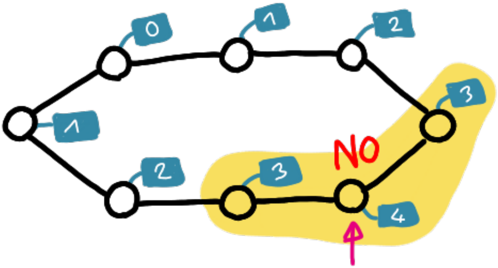
1. Check degree 1 or 2
2. Interpret labels as distances to a root and check consistency.



Example 2

Problem : Is the graph a path ?
Algorithm :

1. Check degree 1 or 2
2. Interpret labels as distances to a root and check consistency.



More on certification

Where it comes from : self-stabilizing algorithms

How to measure performance : The certification size, *i.e.* the minimum size for the certificates for recognizing X.

1. Trees (and paths) : $\Theta(\log n)$
2. Diameter=3 : $\tilde{\Theta}(n)$
3. Any class : $O(n^2)$
4. Symmetric graphs : $\Theta(n^2)$

[Note : In this talk, identifiers are "hidden".]

Certifying planarity

Theorem :

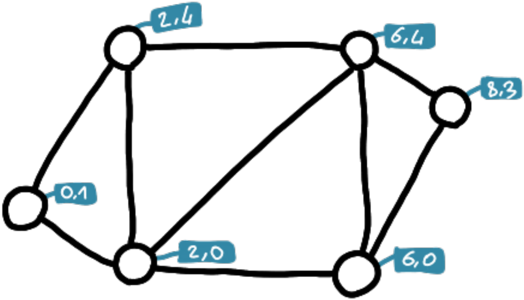
The certification size for planarity is $\Theta(\log n)$.

What follows (only about upper bound) :

- Natural techniques that do not work.
- Solving a special case.
- Going back to the general case.

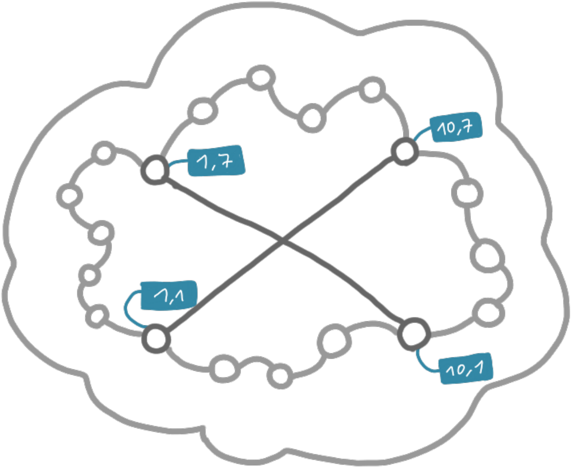
Things that do not work

- Coordinates
- Face numbering



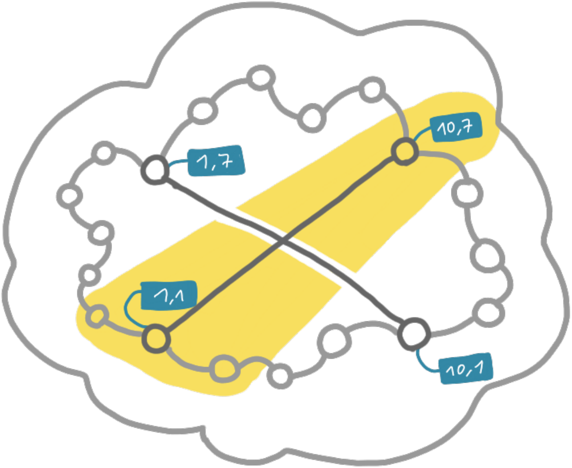
Things that do not work

- Coordinates
- Face numbering



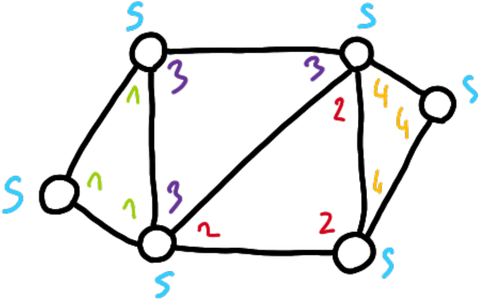
Things that do not work

- Coordinates
- Face numbering



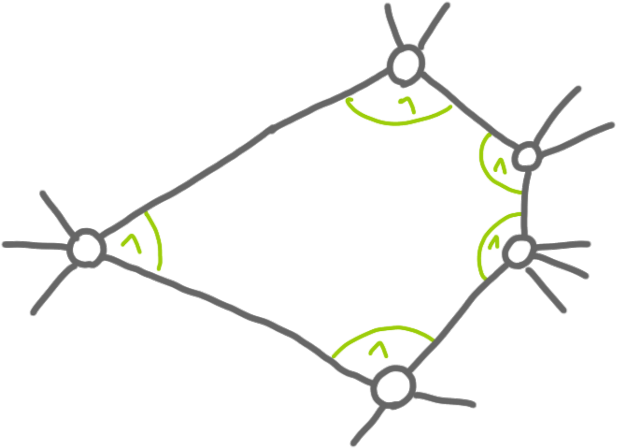
Things that do not work

- Coordinates
- Face numbering



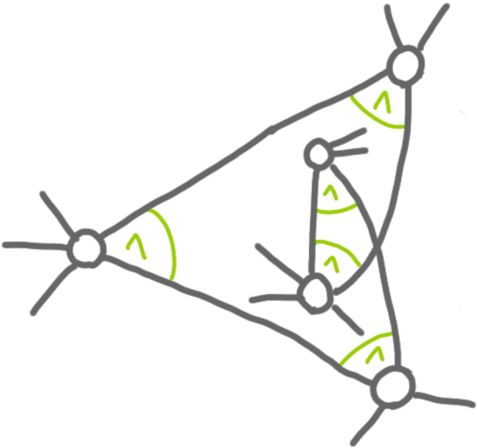
Things that do not work

- Coordinates
- Face numbering



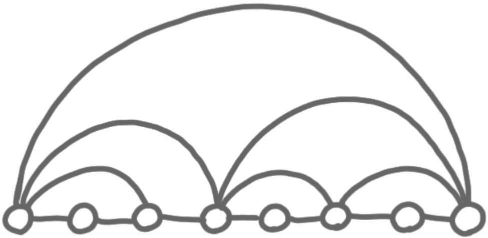
Things that do not work

- Coordinates
- Face numbering



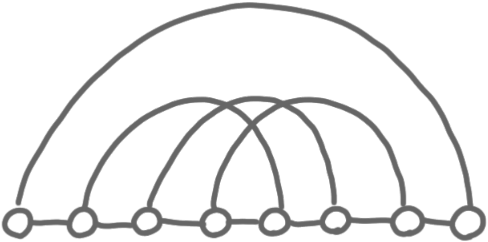
Path-outerplanar case

Definition :
Path-outerplanar = a path with non-crossing edges on top.



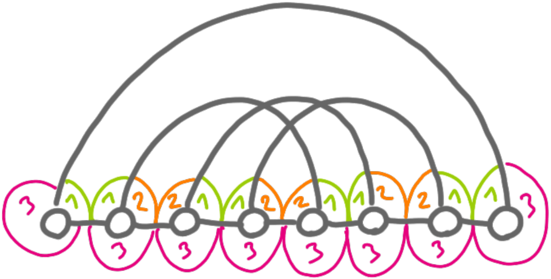
Path-outerplanar case

Definition :
Path-outerplanar = a path with non-crossing edges on top.



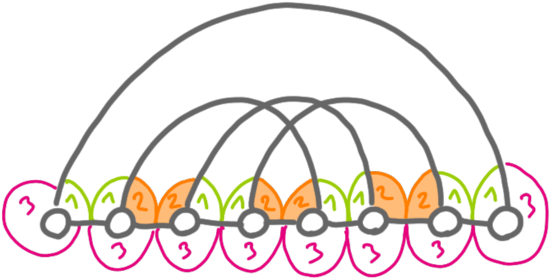
Path-outerplanar case

Definition :
Path-outerplanar = a path with non-crossing edges on top.



Path-outerplanar case

Definition :
Path-outerplanar = a path with non-crossing edges on top.

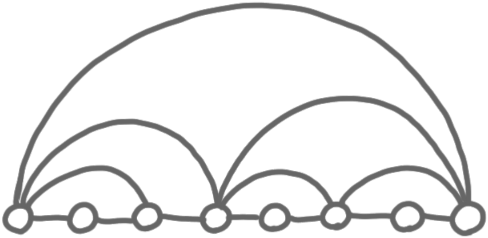


Path-outerplanar case

Expected certificates :

- rank + certification of rank
- name of the edge "above" the node

Algorithm : check "consistency".

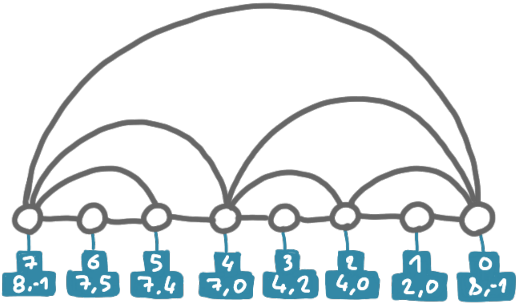


Path-outerplanar case

Expected certificates :

- rank + certification of rank
- name of the edge “above” the node

Algorithm : check “consistency”.

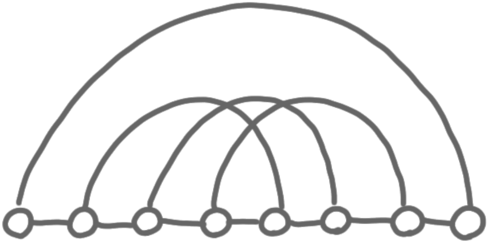


Path-outerplanar case

Expected certificates :

- rank + certification of rank
- name of the edge "above" the node

Algorithm : check "consistency".

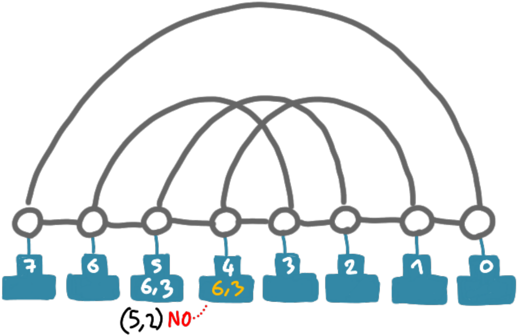


Path-outerplanar case

Expected certificates :

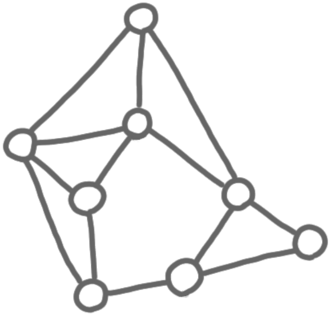
- rank + certification of rank
- name of the edge "above" the node

Algorithm : check "consistency".



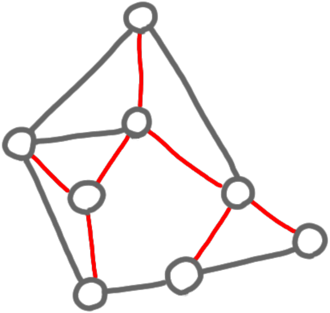
General case

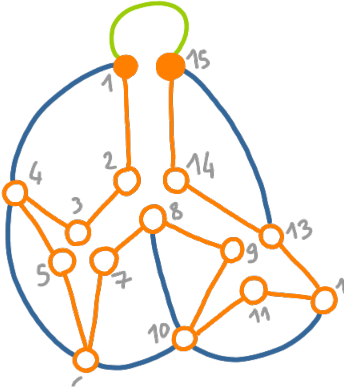
(Certified) transformation from a general planar graph to a path-outerplanar graph.



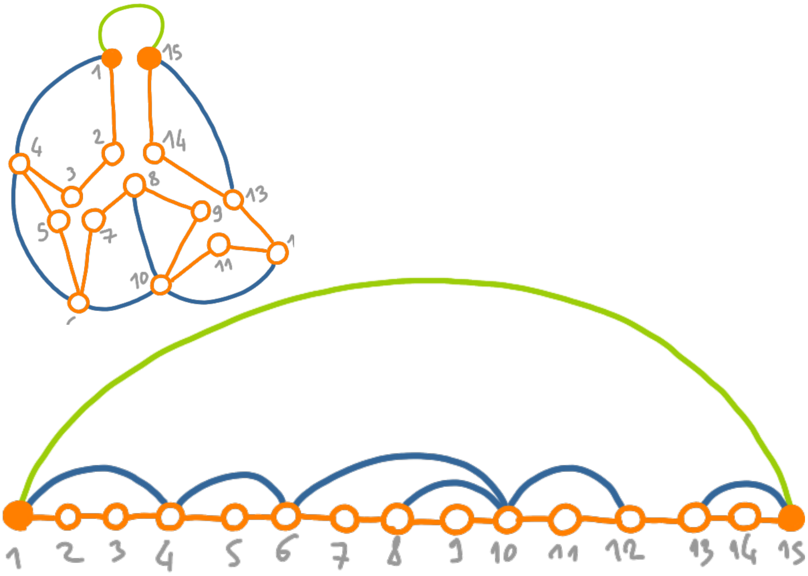
General case

(Certified) transformation from a general planar graph to a path-outerplanar graph.





General case



Conclusion

Not in this talk :

- Local certification beyond graph classes.
- Parts of the scheme (e.g. checking the transformation)
- Lower bounds (that are actually more general)

Next step : bounded genus graphs (tougher than expected) and minor-free graphs (probably wild).