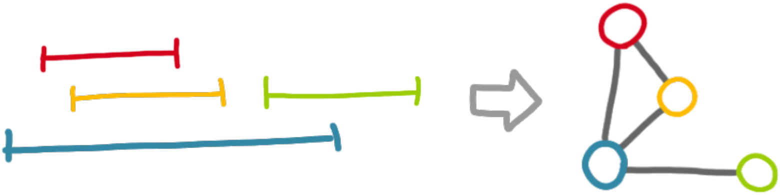
Graph classes and forbidden patterns on three and four vertices

Laurent Feuilloley and Michel Habib

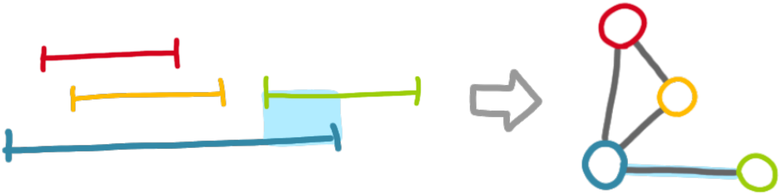
Based on
Graph classes and forbidden patterns on three vertices
(to appear in SIDMA)
and on on-going work.

IRIF Graph Seminar · 24th November 2020

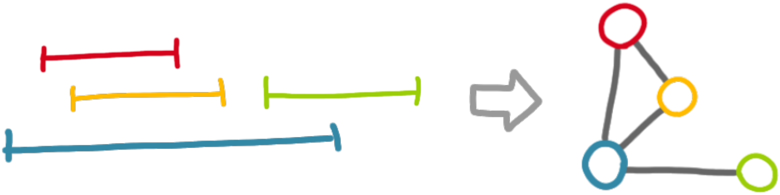
Interval graphs : geometry



Interval graphs : geometry



Interval graphs : geometry



Definition : A graph is an interval graph if it is the intersection graph of a set of intervals.

Interval graphs : geometry

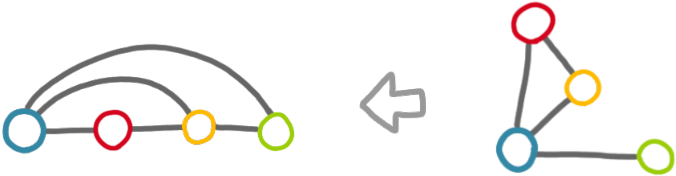
Definition : A graph is an interval graph if it is the intersection graph of a set of intervals.

Interval graphs : with an ordering

Characterization : A graph is an interval graph if and only if, there exists an ordering of its vertices such that for every $u < v < w$, if (u, w) is an edge then (u, v) is also an edge.

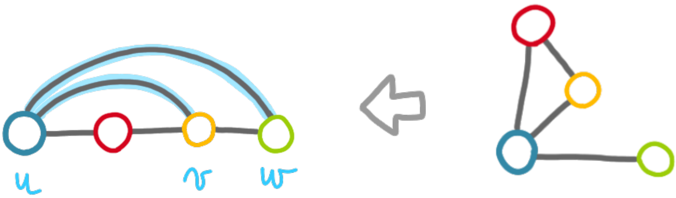
Interval graphs : with an ordering

Characterization : A graph is an interval graph if and only if, there exists an ordering of its vertices such that for every $u < v < w$, if (u, w) is an edge then (u, v) is also an edge.



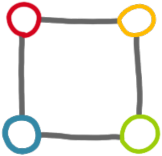
Interval graphs : with an ordering

Characterization : A graph is an interval graph if and only if, there exists an ordering of its vertices such that for every $u < v < w$, if (u, w) is an edge then (u, v) is also an edge.



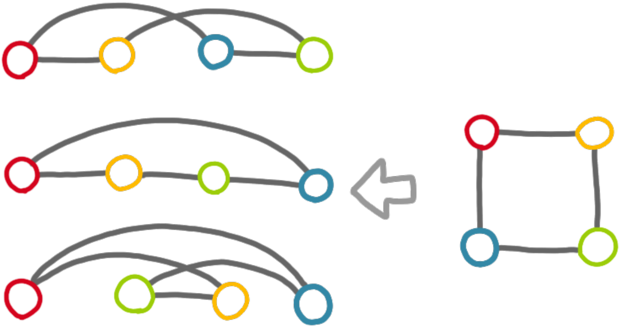
Interval graphs : with an ordering

Characterization : A graph is an interval graph if and only if, there exists an ordering of its vertices such that for every $u < v < w$, if (u, w) is an edge then (u, v) is also an edge.



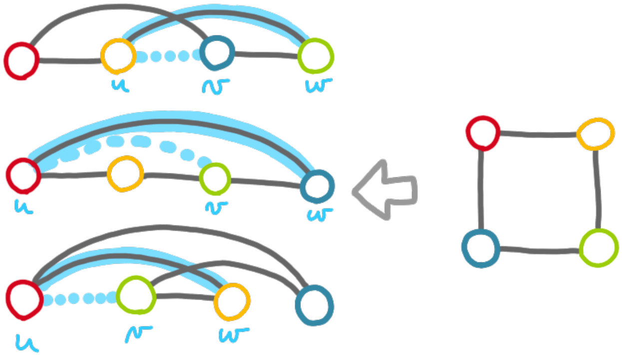
Interval graphs : with an ordering

Characterization : A graph is an interval graph if and only if, there exists an ordering of its vertices such that for every $u < v < w$, if (u, w) is an edge then (u, v) is also an edge.



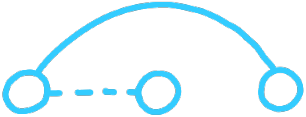
Interval graphs : with an ordering

Characterization : A graph is an interval graph if and only if, there exists an ordering of its vertices such that for every $u < v < w$, if (u, w) is an edge then (u, v) is also an edge.



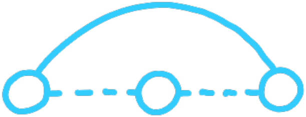
Pattern characterization

Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following pattern does not appear :



Pattern characterization

Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following pattern does not appear :



Pattern characterization

Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following pattern does not appear :

Pattern characterization

Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following pattern does not appear :

INTERVAL

Pattern characterization

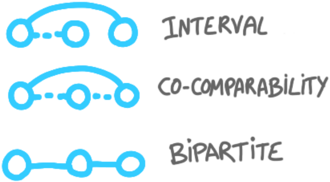
Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following pattern does not appear :

 INTERVAL

 CO-COMPARABILITY

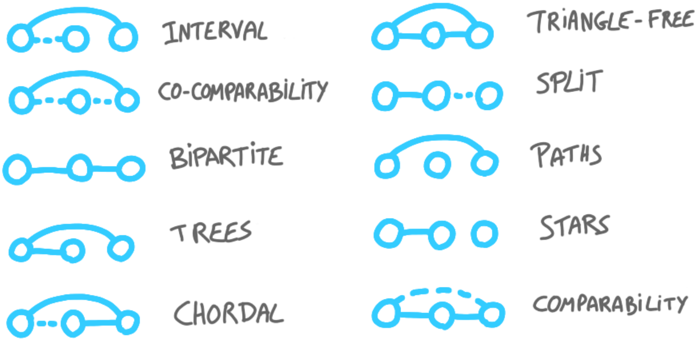
Pattern characterization

Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following pattern does not appear :



Pattern characterization

Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following pattern does not appear :



Already noted by Skrien in 82 and Damashke in 90.

Example : bipartite graphs

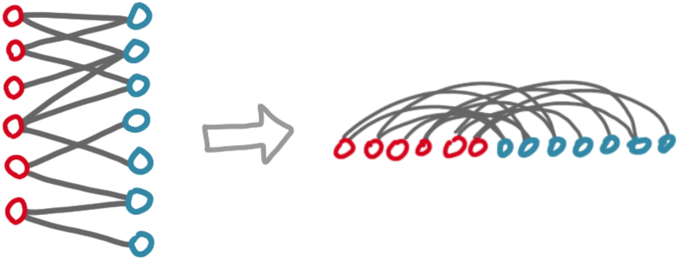
DEFINITION: A GRAPH IS BIPARTITE IF IT CAN BE SPLIT INTO TWO INDEPENDENT SETS

CHARACTERIZATION: A GRAPH IS BIPARTITE IF THERE EXISTS A VERTEX ORDERING WITHOUT:

Example : bipartite graphs

DEFINITION: A GRAPH IS BIPARTITE IF IT CAN BE SPLIT INTO TWO INDEPENDENT SETS

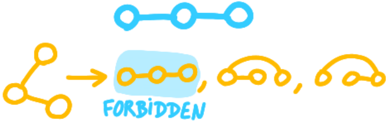
CHARACTERIZATION: A GRAPH IS BIPARTITE IF THERE EXISTS A VERTEX ORDERING WITHOUT:



Example : bipartite graphs

DEFINITION: A GRAPH IS BIPARTITE IF IT CAN BE SPLIT INTO TWO INDEPENDENT SETS

CHARACTERIZATION: A GRAPH IS BIPARTITE IF THERE EXISTS A VERTEX ORDERING WITHOUT:



Example : bipartite graphs

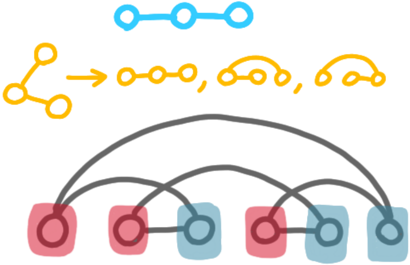
DEFINITION: A GRAPH IS BIPARTITE IF IT CAN BE SPLIT INTO TWO INDEPENDENT SETS

CHARACTERIZATION: A GRAPH IS BIPARTITE IF THERE EXISTS A VERTEX ORDERING WITHOUT:

Example : bipartite graphs

DEFINITION: A GRAPH IS BIPARTITE IF IT CAN BE SPLIT INTO TWO INDEPENDENT SETS

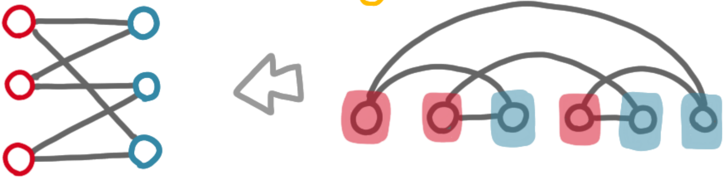
CHARACTERIZATION: A GRAPH IS BIPARTITE IF THERE EXISTS A VERTEX ORDERING WITHOUT:



Example : bipartite graphs

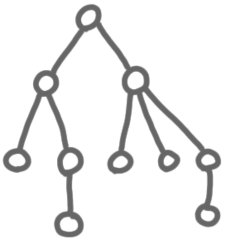
DEFINITION: A GRAPH IS BIPARTITE IF IT CAN BE SPLIT INTO TWO INDEPENDENT SETS

CHARACTERIZATION: A GRAPH IS BIPARTITE IF THERE EXISTS A VERTEX ORDERING WITHOUT:



Example : trees

DEFINITION: A GRAPH IS A TREE IF IT IS ACYCLIC (*)



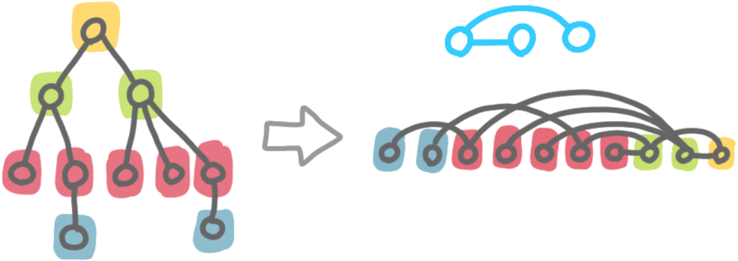
CHARACTERIZATION: A GRAPH IS A TREE IF THERE EXISTS A VERTEX ORDERING WITHOUT:

(*) LET'S NOT WORRY ABOUT CONNECTIVITY

Example : trees

DEFINITION: A GRAPH IS A TREE IF IT IS ACYCLIC (*)

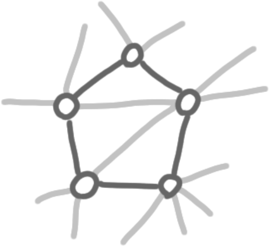
CHARACTERIZATION: A GRAPH IS A TREE IF THERE EXISTS A VERTEX ORDERING WITHOUT:



(*) LET'S NOT WORRY ABOUT CONNECTIVITY

Example : trees

DEFINITION: A GRAPH IS A TREE IF IT IS ACYCLIC (*)



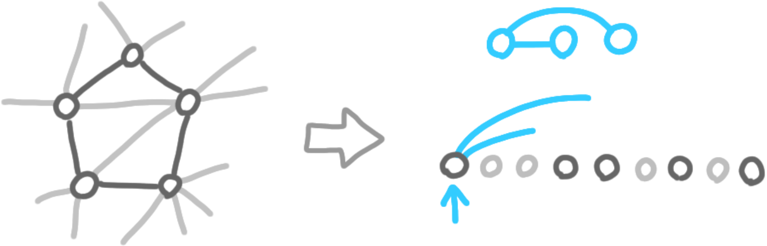
CHARACTERIZATION: A GRAPH IS A TREE IF THERE EXISTS A VERTEX ORDERING WITHOUT:

(*) LET'S NOT WORRY ABOUT CONNECTIVITY

Example : trees

DEFINITION : A GRAPH IS A TREE IF IT IS ACYCLIC (*)

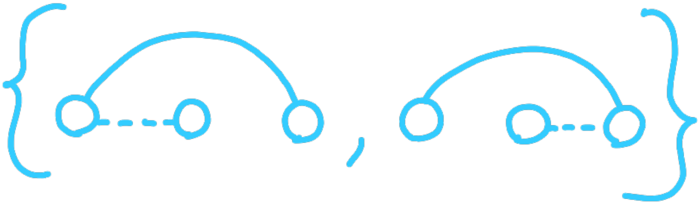
CHARACTERIZATION: A GRAPH IS A TREE IF THERE EXISTS A VERTEX ORDERING WITHOUT:



(*) LET'S NOT WORRY ABOUT CONNECTIVITY

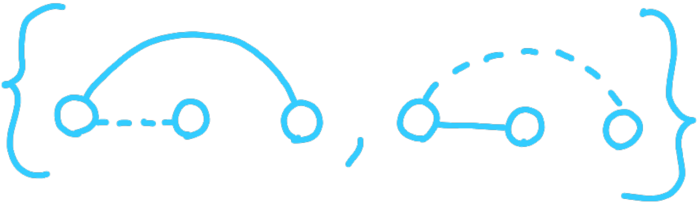
Pattern characterization

Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following **patterns** do not appear :



Pattern characterization

Characterization : A graph is a XXX if and only if, there exists an ordering of its vertices such that the following **patterns** do not appear :



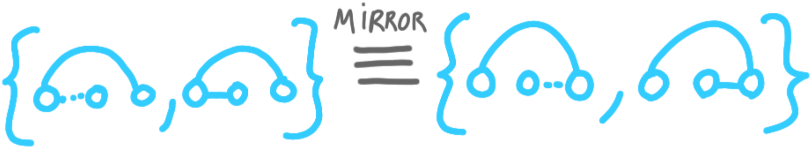
Theorem

Theorem : Up to a few simple operations, the non-trivial classes defined by a set of pattern (on three nodes) are :

1. forests
2. linear forests
3. stars
4. interval
5. split
6. bipartite
7. chordal
8. comparability
9. triangle-free
10. permutation
11. threshold
12. proper interval
13. caterpillar
14. trivially perfect
15. bipartite chain
16. 2-star
17. 1-split
18. augmented clique
19. bipartite permutation
20. triangle-free $\cap$ co-chordal
21. clique
22. complete bipartite

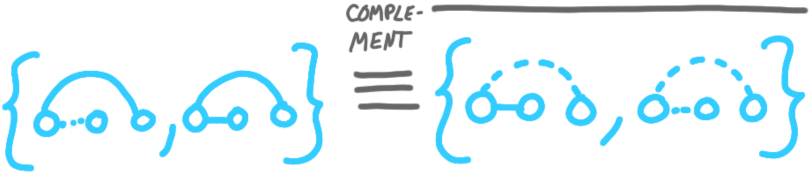
Structure

- Mirror patterns



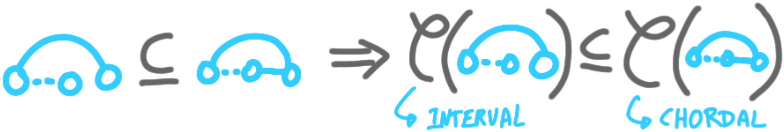
Structure

- Mirror patterns
- Complementary patterns



Structure

- Mirror patterns
- Complementary patterns
- Inclusions of patterns $\rightarrow$ classes inclusions

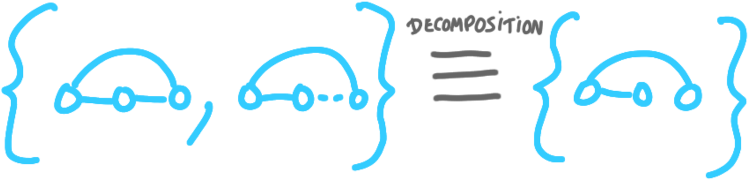


Structure

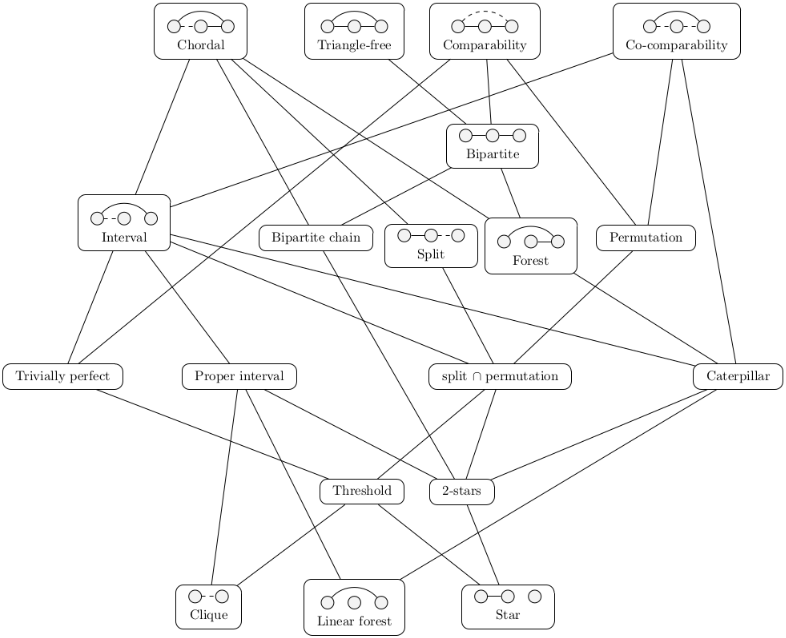
- Mirror patterns
- Complementary patterns
- Inclusions of patterns $\rightarrow$ classes inclusions
- Intersections

Structure

- Mirror patterns
- Complementary patterns
- Inclusions of patterns $\rightarrow$ classes inclusions
- Intersections
- "Splitting patterns"



Structure



Chordal
Triangle-free
Comparability
Co-comparability
Bipartite
Interval
Bipartite chain
Split
Forest
Permutation
Trivially perfect
Proper interval
split ∩ permutation
Caterpillar
Threshold
2-stars
Clique
Linear forest
Star

Complexity and algorithms

An ordering can be checked in polytime $\rightarrow$ Recognition is in NP.

Theorem (Hell, Mohar and Rafiey) : Every class defined by a set of patterns on three nodes can be recognized in time $O(n^3)$.

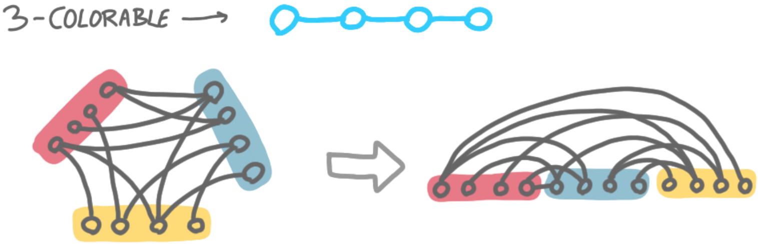
New theorem : Every class defined by patterns on three nodes can be recognized in **linear time** except two of them (in time $O(n^{2,37})$), and mostly thanks to **graph traversals**.

What about 4 vertices ?

- A lot more cases, much less is known.

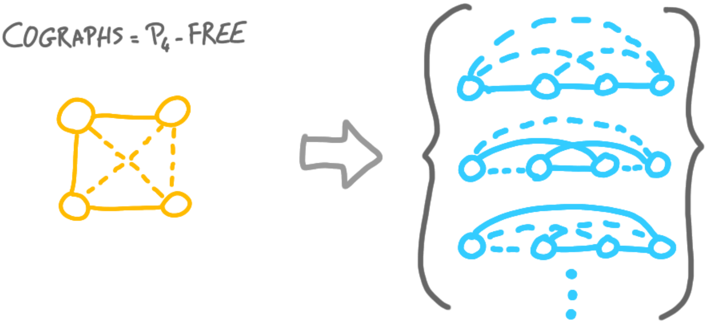
What about 4 vertices ?

- A lot more cases, much less is known.
- 3-colorable graphs, with one pattern on 4 nodes.
  $\rightarrow$ NP-complete recognition.



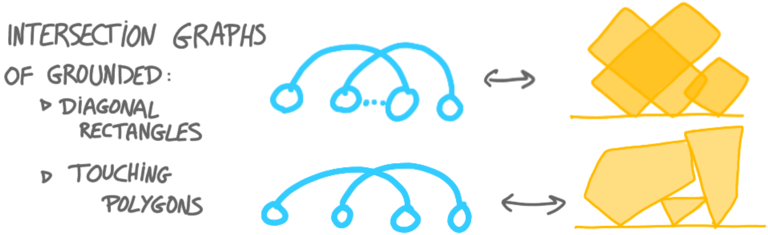
What about 4 vertices ?

- A lot more cases, much less is known.
- 3-colorable graphs, with one pattern on 4 nodes.
  $\rightarrow$ NP-complete recognition.
- Classes related to P_4 : cographs, trivially perfect ($\rightarrow$ general transformation)



What about 4 vertices ?

- A lot more cases, much less is known.
- 3-colorable graphs, with one pattern on 4 nodes.
  $\rightarrow$ NP-complete recognition.
- Classes related to P_4 : cographs, trivially perfect ($\rightarrow$ general transformation)
- Grounded intersection graphs



Grounded intersection graphs

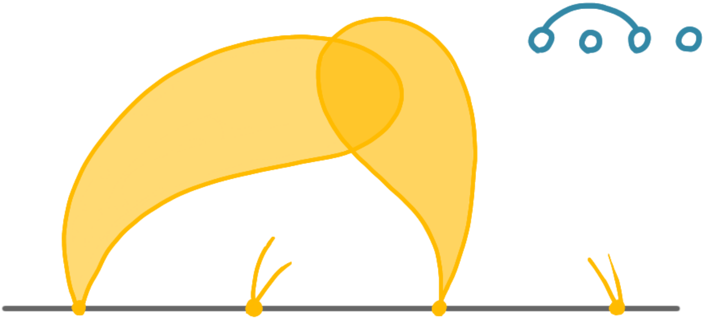
GROUNDED SHAPES $\subseteq \mathcal{C}\Big($  $\Big)$

Grounded intersection graphs

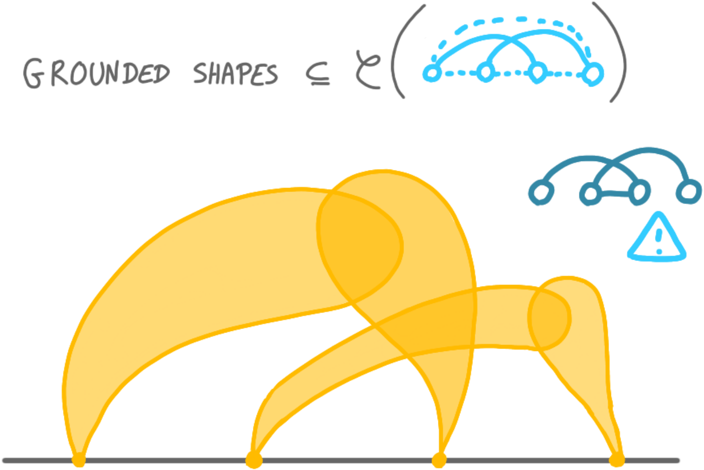
GROUNDED SHAPES $\subseteq \mathcal{C}\Big($  $\Big)$

Grounded intersection graphs

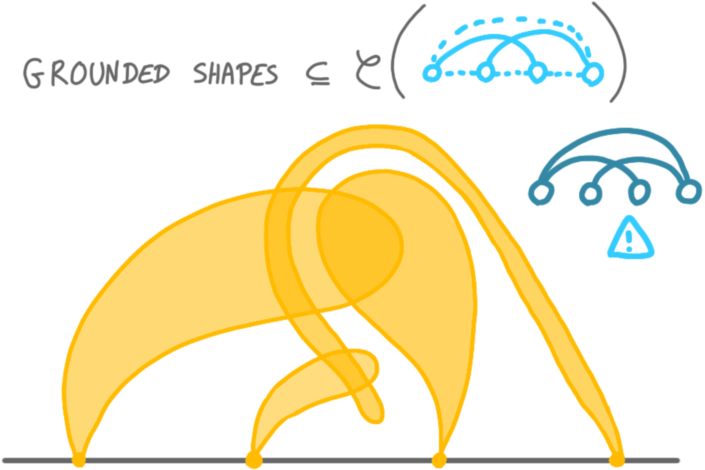
GROUNDED SHAPES $\subseteq \mathcal{C}\Big($  $\Big)$



Grounded intersection graphs



Grounded intersection graphs



Open problems

A few concrete open problems :

- Clarify the interplay between grounded intersection model and patterns.
- Complexity of recognition of grounded rectangle graphs : P or NP ?
- Find a criterion for deciding the complexity of the class based on its patterns.