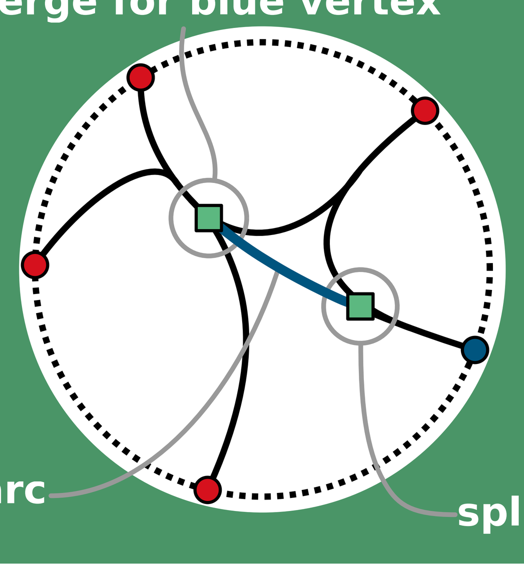
Towards Characterizing Strict Outerconfluent Graphs

Fabian Klute · Martin Nöllenburg

Definition ([1,5])

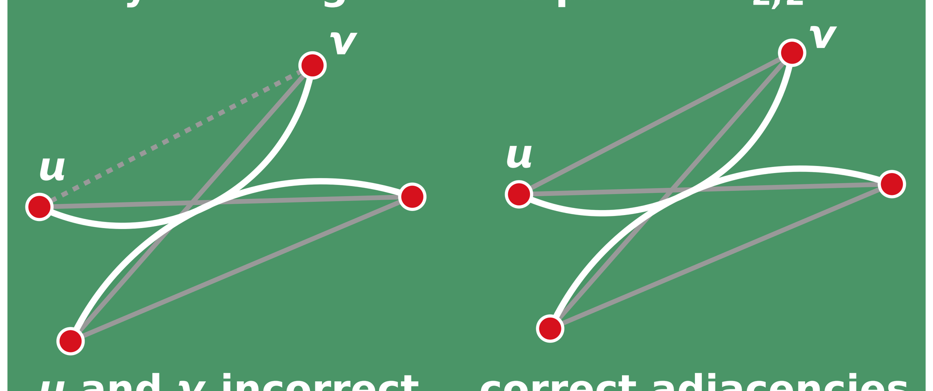
Given graph $G = (V, E)$, we search a confluent drawing D s.t.:



- Vertices $v \in V$ are mapped to points
- Edges $(u, v) \in E$ are mapped to smooth curves between vertices
- A path consists of arcs and junction points connecting them
- A drawing D is strict if there is exactly one smooth curve per edge $(u, v) \in E$
- D is outerconfluent if all vertices are adjacent to the outer face

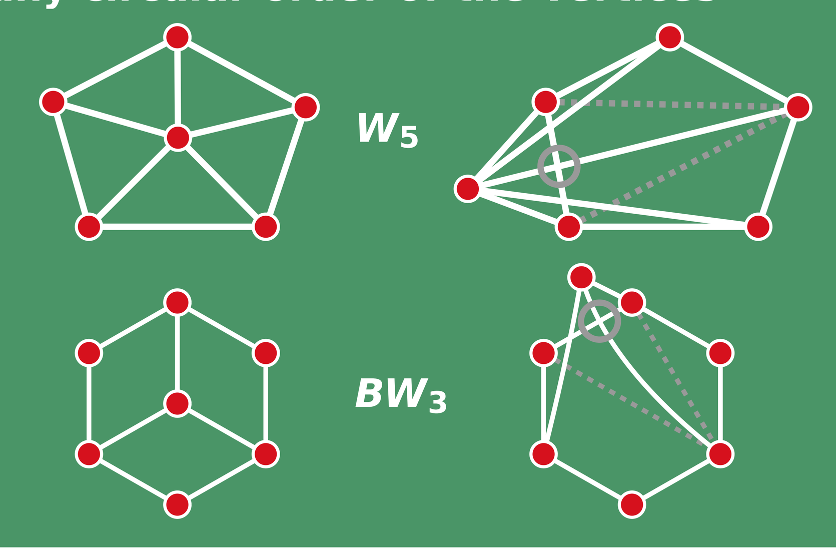
Represented Crossings

Draw graph as circular layout
Crossings determined by vertex order
Every crossing must be part of $K_{2,2}$



Two graphs with no SOC drawing

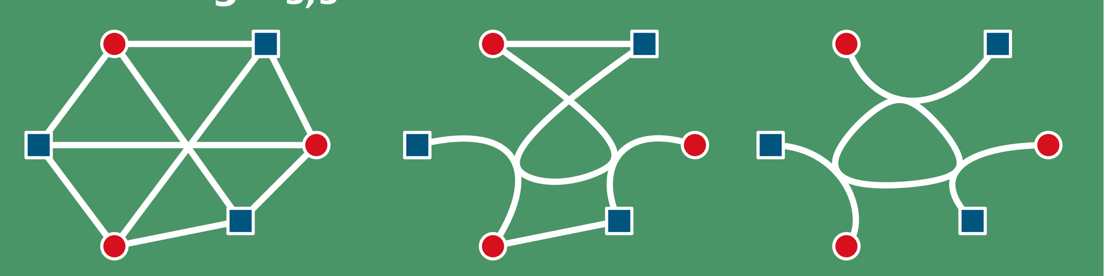
We find a crossing, not represented in any circular order of the vertices



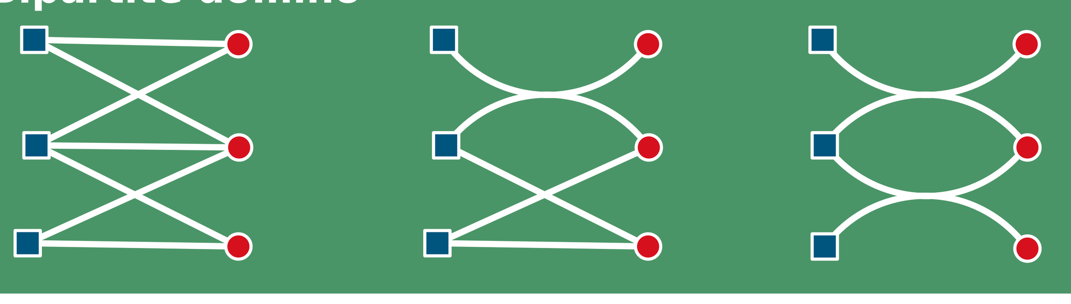
Forbidden Orders

These two orders have no SOC drawing, although represented:

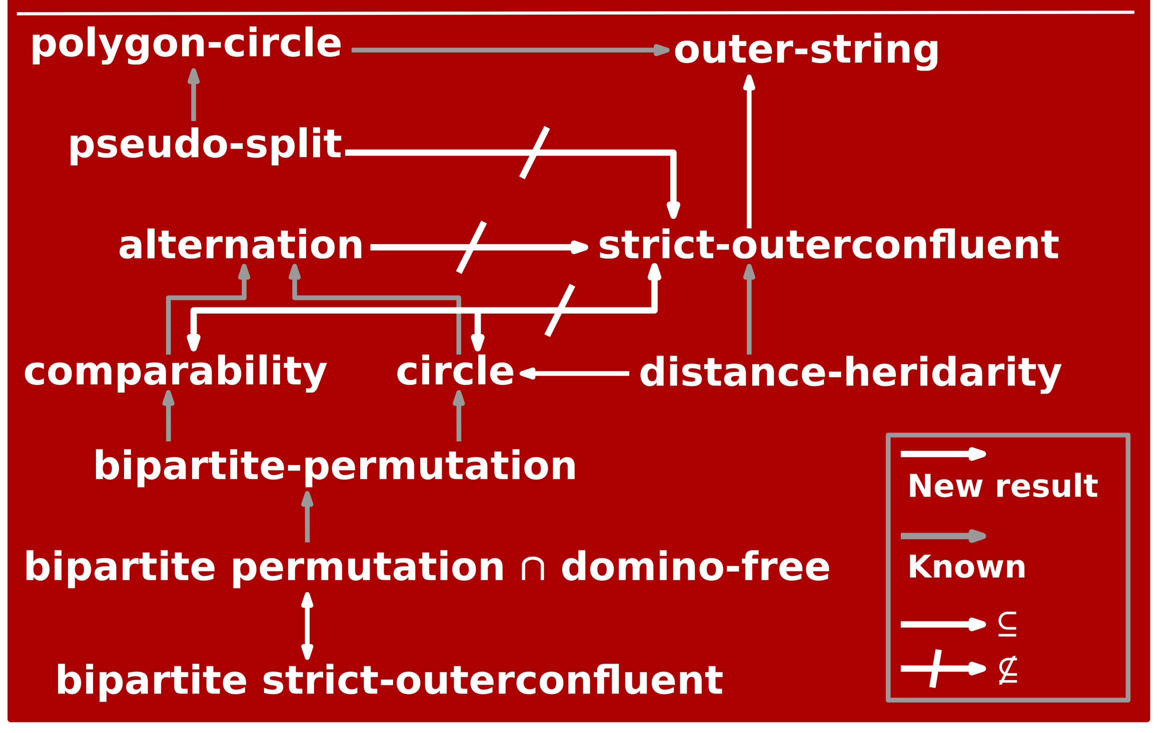
Alternating $K_{3,3}$



Bipartite domino

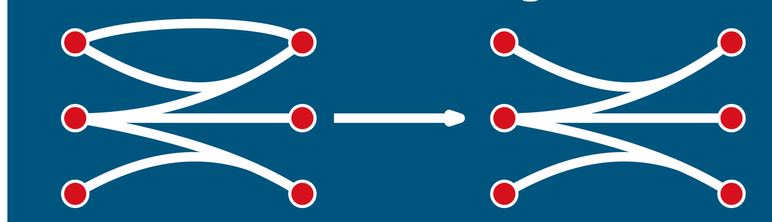


Results at a Glance

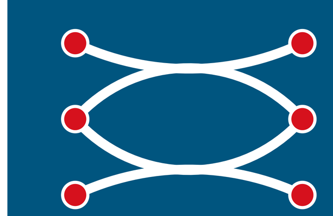


Bipartite SOC

Use Hui's algorithm [2] to get a bipartite outerconfluent drawing



Eliminate non-strict paths when possible



Only Domino graph can not be eliminated

⇒ bipartite-permutation ∩ domino-free ⊆ SOC

Infer graph from bipartite SOC drawing

This graph is a bipartite permutation graph [2]

Domino is not representable as bipartite SOC ⇒ domino-free

⇒ **Bipartite SOC ⊆ bipartite permutation graph ∩ domino-free**

SOC ⊂ outer-string

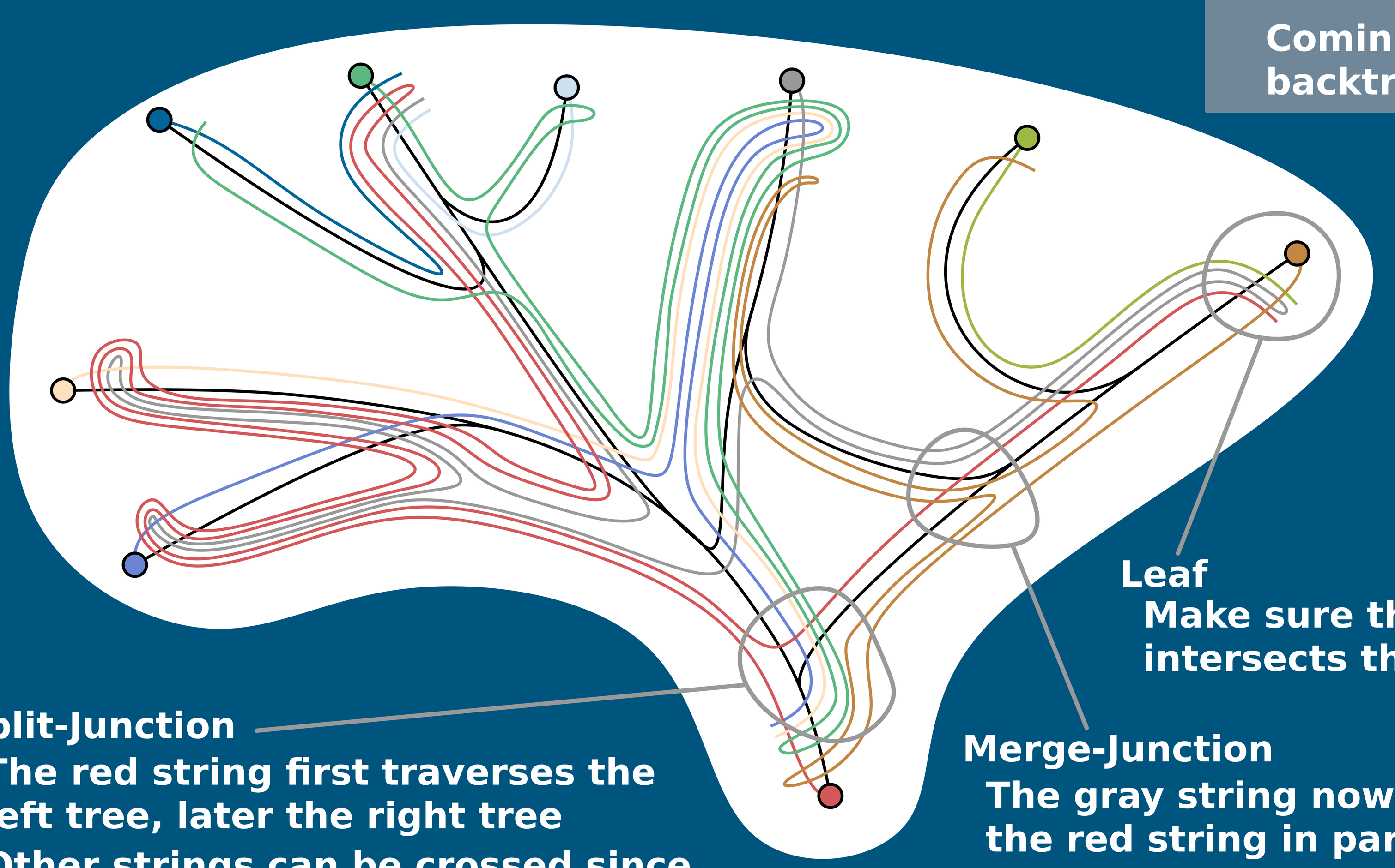
Construct a tree of junctions with respect to one vertex

Build a string for every node in the drawing

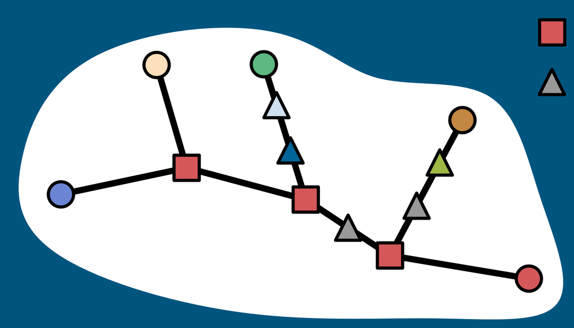
Construct a string representation from individual strings

Building one string

- Traverse tree in left-first DFS order
- Make clockwise U-turn at leaf and backtrack
- At split-junction
  - Coming from left subtree: cross arc and descend into right subtree
  - Coming from right subtree: cross arc and backtrack to split-junction



The tree of the red node



Leaf
Make sure the red string intersects the brown one

Split-Junction
The red string first traverses the left tree, later the right tree
Other strings can be crossed since they see this as merge-junction

Merge-Junction
The gray string now follows the red string in parallel

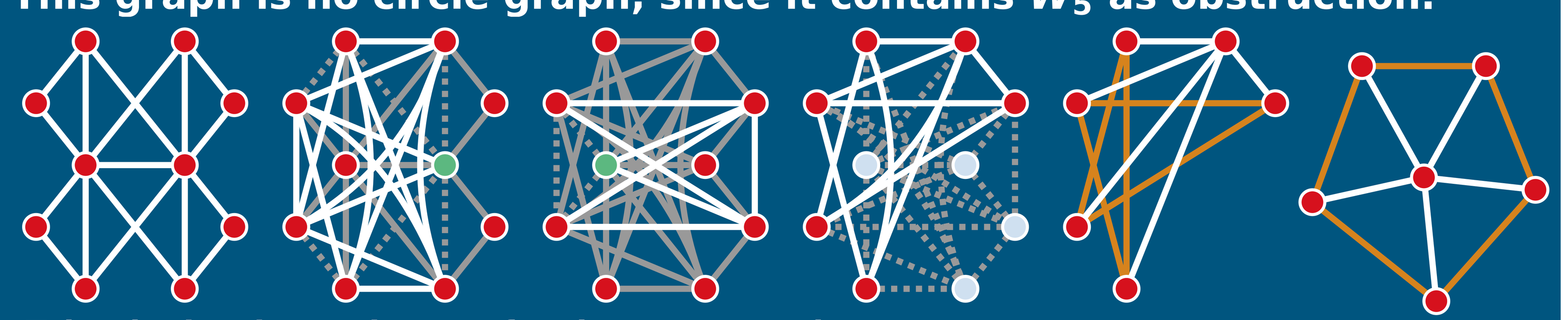
pseudo-split ⊈ SOC

Split W_5 into a clique and a C_5

⇒ pseudo-split ⊈ SOC
pseudo-split ⊂ polygon-circle
⇒ polygon-circle ⊈ SOC

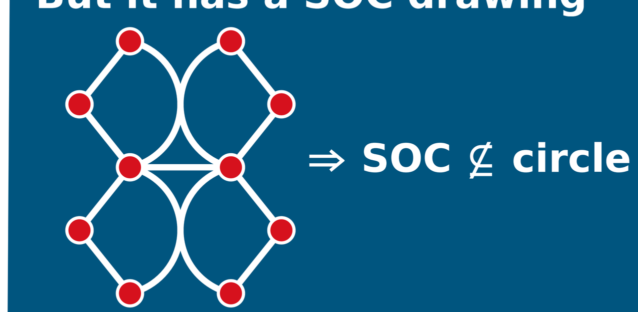
SOC ⊈ circle ⊈ SOC

This graph is no circle graph, since it contains W_5 as obstruction.

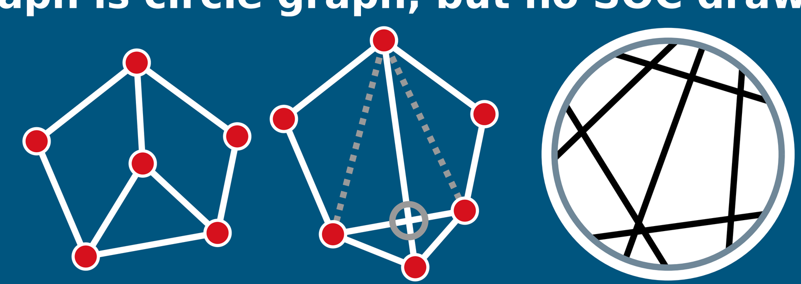


Take the local complement for the green vertices
If W_5 can be found as induced subgraph after sequence of local complements the graph is no circle graph [3]

But it has a SOC drawing



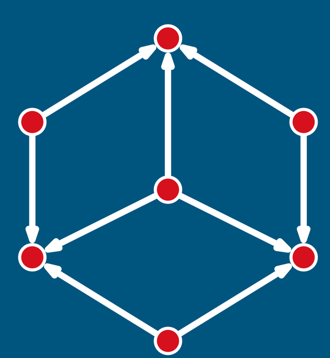
Graph is circle graph, but no SOC drawing



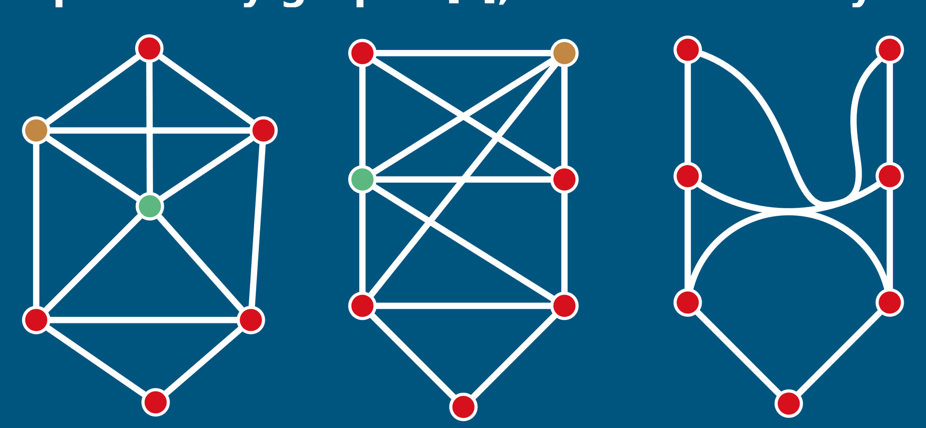
SOC ⊈ comparability ⊈ SOC

A graph is a comparability graph iff it has a transitive orientation

BW_3 has a transitive orientation and is hence a comparability graph



Graph is among forbidden subgraphs of comparability graphs [4], but has SOC layout



Open Questions

The characterization of SOC graphs remains open

Interesting questions:

- Is every SOC graph an alternation/circle-polygon graph?
- Distance-Heridarity graphs have rankwidth 1 and are in SOC, what about rankwidth 2,3,...
- Can we draw every permutation graph as SOC?

alternation ⊈ SOC

circle and comparability are subclasses of alternation

circle ⊈ SOC, circle ⊆ alternation
⇒ alternation ⊈ SOC

comp ⊈ SOC, comp⊆ alternation
⇒ alternation ⊈ SOC

Graph Glossary

Comparability graph iff it has transitive orientation

Circle graph iff has intersection model of chords in a circle

Permutation graph iff it has an intersection model of lines between two parallels

Outer-string graph iff has intersection model of curves in a circle with one endpoint on the circle

Polygon-circle iff is the intersection model of polygons inscribed in a circle

Alternation graph iff it has semi-transitive orientation

Pseudo-Split iff vertices can be partitioned into a complete graph, a C_5 and an independent set. Complete graph adjacent to C_5, independent set not adjacent to C_5

[1] Eppstein, D., Holten, D., Löffler, M., Nöllenburg, M., Speckmann, B., Verbeek, K.: Strict confluent drawing. Journal of Computational Geometry 7(1), 22–46 (2016)
[2] Hui, P., Pelsmajer, M.J., Schaefer, M., Stefankovic, D.: Train tracks and confluent drawings. Algorithmica 47(4), 465–479 (2007)
[3] A. Bouchet. Circle graph obstructions. Journal of Combinatorial Theory, Series B, 60(1):107–144, 1994.
[4] T. Gallai. Transitiv orientierbare Graphen. Acta Mathematica Hungarica, 18(1- 2):25–66, 1967.
[5] M. Dickerson et al. Confluent drawings: visualizing non-planar diagrams in a planar way. In GD'03, volume 2912 of LNCS, pages 1–12. Springer, 2003.

TU WIEN
ac ALGORITHMS AND COMPLEXITY GROUP