Towards Characterizing Strict Outerconfluent Graphs

Fabian Klute and Martin Nöllenburg

Algorithms and Complexity Group, TU Wien, Vienna, Austria

Confluent drawings of graphs are geometric representations in the plane, in which vertices are mapped to points, but edges are not drawn as individually distinguishable geometric objects. Instead, an edge is represented by the presence of a smooth curve between two vertices in a system of arcs and junctions.

More formally, a confluent drawing D of a graph $G = (V, E)$ consists of a set of points representing the vertices, a set of junction points, and a set of smooth arcs, such that each arc starts and ends at a vertex point or a junction, no two arcs intersect (except at common endpoints), and all arcs meeting in a junction share the same tangent line in the junction point. There is an edge $(u, v) \in E$ if and only if there is a smooth path from u to v in D that does not pass through any other vertex.

Confluent drawings were introduced by Dickerson et al. [1], who identified classes of graphs that admit or that do not admit confluent drawings. Later, variations such as strong and tree confluency [6], as well as Δ-confluency [2] were introduced. Confluent drawings have further been used for layered drawings [3] and for drawing Hasse diagrams [5]. The complexity of the recognition problem for graphs that admit a confluent drawing remains open.

Eppstein et al. [4] defined *strict* confluent drawings, in which every edge of the graph must be represented by a unique smooth path. They showed that for general graphs it is NP-complete to decide whether a strict confluent drawing exists. A strict confluent drawing is called *strict outerconfluent* if all vertices lie on the boundary of a (topological) disk that contains the strict confluent drawing. For a given cyclic vertex order, Eppstein et al. [4] presented a constructive poly-time algorithm for testing the existence of a strict outerconfluent drawing. Without a given vertex order the recognition complexity as well as a characterization of the graphs admitting such drawings remained open. We present first results towards characterizing the strict outerconfluent (SOC) graphs by examining potential sub- and super-classes of SOC graphs. For definitions of the used graph classes we refer to `www.graphclasses.org`.

If we draw a graph G as a traditional circular drawing with straight-line edges, then all the crossings are determined by the order of the vertices alone. We can replace a crossing by a confluent junction if the two edges forming the crossing are part of a $K_{2,2}$. We call such a crossing *represented*. It is clear that a graph can only have a strict outerconfluent drawing if it has a circular layout with all crossings represented. This is not sufficient though, as there are such graphs that have no strict outerconfluent drawing. We obtain two 6-vertex obstructions for strict outerconfluent drawings, namely a $K_{3,3}$ with an alternating vertex order and a domino graph (two four-cycles sharing an edge) in bipartite order.

Our next result concerns bipartite drawings. Let $G = (X, Y, E)$ be a bipartite graph with vertex sets X and Y. We call a strict outerconfluent drawing D a *bipartite strict outerconfluent drawing* if the nodes can be partitioned into two independent sets, such that each set is consecutive on the boundary of the topological disk. Hui et al. [6] showed that the bipartite outerconfluent graphs are exactly the bipartite permutation graphs. We show that the (bipartite-permutation $\cap$ domino-free) graphs are exactly the bipartite strict outerconfluent graphs. The proof uses the drawing algorithm by Hui et al. to obtain a confluent bipartite drawing, which is non-strict if and only if a domino is present.

On the other hand we show that circle and comparability graphs are neither sub- nor superclasses of the SOC graphs and the alternation and circle-polygon graphs are no sub-classes of them. All the results can be shown via counterexamples, mostly using the wheel on six vertices and the so-called BW_3 graph, which both have no SOC drawing.

Finally our main result shows an interesting superclass of SOC graphs. The class of *outer-string* graphs contains all graphs $G = (V, E)$ which can be represented by an intersection model of curves in a disk with one end-point on the disk's boundary. We show that SOC graphs are outer-string graphs. The inclusion is proper, because not every circle-polygon graph is an SOC graph, but every circle-polygon graph is an outer-string graph.

Let D be a strict outerconfluent drawing. To get an outer-string representation of the corresponding graph G_D we need to find for every vertex v in G_D a string starting at the node representing v in D and intersecting only strings representing adjacent vertices in G_D. We do this by exploiting the tree structure we get for one node in D, when looking at all the junctions and other nodes which can be reached from it via smooth paths. We call a junction j *split-junction*, if the path coming from v separates at j into two paths and *merge-junction* if another path fuses with it at j. One string is then constructed as follows:

- Start from a node and traverse its tree in left-first DFS order
- At leaf, make a clockwise U-turn and backtrack to the previous split-junction.
- At split-junction:
  - coming from the left subtree: cross the arc from the left subtree at the junction and descend into the right subtree
  - coming from the right subtree: cross the arc to the left subtree and backtrack along the existing string to the previous split-junction

To find the complete outer-string representation of G_D we have to combine all these strings for nodes in D. We distinguish three cases, two of which are straightforward. If two nodes are connected by a path we have to guarantee that the two strings intersect at least once, which can be done at the leaves. The second one considers two nodes without a path connecting them and the two trees are independent, i.e., not sharing a junction. Then the strings are independent by construction as well. Finally if the trees share junctions, then these can be only merge-junctions. The key observation here is that at most two merge-junctions can be shared by two nodes without a connecting path in D.

References

1. Dickerson, M., Eppstein, D., Goodrich, M.T., Meng, J.Y.: Confluent drawings: Visualizing non-planar diagrams in a planar way. J. Graph Algorithms and Applications 9(1), 31–52 (2005)
2. Eppstein, D., Goodrich, M.T., Meng, J.Y.: Delta-confluent drawings. In: Healy, P., Nikolov, N.S. (eds.) Graph Drawing (GD'05). LNCS, vol. 3843, pp. 165–176. Springer (2006)
3. Eppstein, D., Goodrich, M.T., Meng, J.Y.: Confluent layered drawings. Algorithmica 47, 439–452 (2007)
4. Eppstein, D., Holten, D., Löffler, M., Nöllenburg, M., Speckmann, B., Verbeek, K.: Strict confluent drawing. Journal of Computational Geometry 7(1), 22–46 (2016)
5. Eppstein, D., Simons, J.A.: Confluent hasse diagrams. Journal of Graph Algorithms and Applications 17(7), 689–710 (2013)
6. Hui, P., Pelsmajer, M.J., Schaefer, M., Stefankovic, D.: Train tracks and confluent drawings. Algorithmica 47(4), 465–479 (2007)