STRICT CONFLUENT DRAWING*

David Eppstein,[†] *Danny Holten,*[‡] *Maarten Löffler,*[§] *Martin Nöllenburg,*[¶] *Bettina Speckmann,*[‖] *and Kevin Verbeek*[‖]

Abstract. We define *strict confluent drawing*, a form of confluent drawing in which the existence of an edge is indicated by the presence of a smooth path through a system of arcs and junctions (without crossings), and in which such a path, if it exists, must be unique. We prove that it is NP-complete to determine whether a given graph has a strict confluent drawing but polynomial to determine whether it has an *outerplanar* strict confluent drawing with a fixed vertex ordering (a drawing within a disk, with the vertices placed in a given order on the boundary).

1 Introduction

Confluent drawing is a style of graph drawing in which edges are not drawn explicitly; instead vertex adjacency is indicated by the existence of a smooth path through a system of arcs and junctions that resemble train tracks. Formally, a confluent drawing may be defined as a collection of *vertices*, *junctions* and *arcs* in the plane, such that all arcs are smooth and start and end at either a junction or a vertex, such that arcs intersect only at their endpoints, and such that all arcs that meet at a junction share the same tangent line there. A confluent drawing D represents a graph G defined as follows: the vertices of G are the vertices of D, and there is an edge between two vertices u and v in G if and only if there exists a smooth path in D from u to v that does not pass through any other vertex. (In some variants of confluent drawing an additional restriction is made that the smooth path may not intersect itself [13]; however, this constraint is not relevant for our work.) Figure 1 illustrates the visual improvement of a simple confluent drawing over the corresponding node-link diagram.

Confluent drawings allow even very dense and non-planar graphs, such as complete graphs and complete bipartite graphs, to be drawn in a planar way [4] (however, note that not all graphs have planar confluent drawings - removing edges may make it harder to draw a graph). Given the abundance of non-planar graphs in many applications and the fact that edge crossings make graph reading tasks difficult [19], confluent drawings are

*A preliminary version of this paper was presented at the 21st International Symposium on Graph Drawing [8].

[†] *Computer Science Department, University of California, Irvine, USA*, eppstein@uci.edu

[‡] Synerscope BV, Eindhoven, the Netherlands, danny.holten@synerscope.com

[§] *Department of Computing and Information Sciences, Utrecht University, the Netherlands*, m.loffler@uu.nl

[¶] *Algorithms and Complexity Group, TU Wien, Vienna, Austria*, noellenburg@ac.tuwien.ac.at

[‖] *Department of Mathematics and Computer Science, TU Eindhoven, the Netherlands*, [b.speckmann|k.a.b.verbeek]@tue.nl

(a) (b)

Figure 1: (a) A graph drawn as a traditional node-link diagram. (b) The same graph, with edges drawn confluently.

an elegant useful way of representing the connectivity information of non-planar graphs without showing explicit edge crossings. Dickerson et al. [4] gave examples of non-planar graph classes that can be drawn confluently, but also graphs that do not have confluent drawings. In a heuristic algorithm, they recursively detect large cliques or bicliques in the graph and replace them with confluent subdrawings. Since its introduction, there has been much subsequent work on confluent drawing. For instance, tree-confluent drawings [13] and Δ-confluent drawings [7] have been characterized, in which the set of arcs of the confluent drawings are tree-like (optionally allowing 3-way Δ junctions). Eppstein et al. [7] showed that Δ-confluent drawings can be drawn in $O(n \log n)$ area based on a hexagonal grid. Hui et al. [13] showed that strong confluent drawings (a subclass of confluent drawings) can be recognized in polynomial time. Hirsch et al. [10] presented a heuristic method for producing confluent drawings from a graph's biclique edge cover graph. Quercini and Ancona [20] described another confluent drawing heuristic based on rectangular duals. Eppstein et al. [6] combined layered drawings [21] with confluent drawings for bicliques between adjacent layers. Confluent layered drawings are not planar, but significantly reduce the number of edge crossings. The complexity of confluent drawing, however, has remained unclear: how difficult is it to determine whether a given graph has a confluent drawing?

Confluent drawings have a certain visual similarity to a graph drawing technique called *edge bundling* [3, 5, 11, 12, 14], in which "similar" edges are routed together in "bundles", but we note that these drawings should be interpreted differently. In particular, sets of edges bundled together form visual junctions, however, interpreting them as confluent junctions can create false adjacencies.

Contribution. In this paper we introduce a subclass of confluent drawings, which we call *strict* confluent drawings. Strict confluent drawings are confluent drawings with the additional restrictions that between any pair of vertices there can be *at most one* smooth path, and there cannot be any paths from a vertex to itself. Figure 2 illustrates the forbidden configurations. To avoid irrelevant components in the drawing, we also require all arcs of the drawing to be part of at least one smooth path representing an edge. We believe that these restrictions may make strict drawings easier to read, by reducing the ambiguity caused by the existence of multiple paths between vertices. In addition, as we show, the assumption of strictness allows us to completely characterize their complexity, the first such characterization for any form of confluence on arbitrary undirected graphs.

We prove the following:

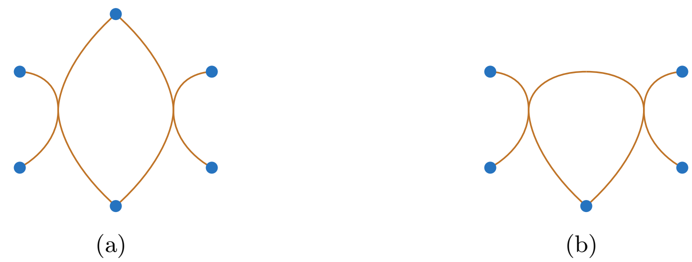


Figure 2: Two examples of non-strict confluent drawings. (a) A drawing with a duplicate path. (b) A drawing with a self-loop.

- It is NP-complete to determine whether a given graph has a strict confluent drawing.
- For a given graph, with a given cyclic ordering of its vertices, there is a polynomial time algorithm to find an *outerplanar* strict confluent drawing, if it exists: this is a drawing in a disk, with the vertices in the given order on the boundary of the disk
- For a graph that has an outerplanar strict confluent drawing, an algorithm based on circle packing can construct such a drawing, in which every arc is drawn using at most two circular arcs.

Figure 3(a) shows an example of an outerplanar strict confluent drawing. It is important to note that an outerplanar strict confluent drawing can be visually reminiscent of outerplanar graph drawings, but the concept is very different in terms of the graph classes having such a drawing; in fact, every outerplanar graph is also outerplanar strict confluent but the reverse is not true. Outerplanar graphs are precisely those planar graphs that have a planar drawing, in which all vertices are incident to the unbounded outer face. An outerplanar strict confluent drawing, generally speaking, is a strict confluent drawing, in which all vertices are incident to the unbounded outer face; junctions can be incident to any face of the drawing. Highly non-planar graphs may have outerplanar strict confluent drawings.

Previous work on *tree-confluent* [13] and *delta-confluent drawings* [7] characterized

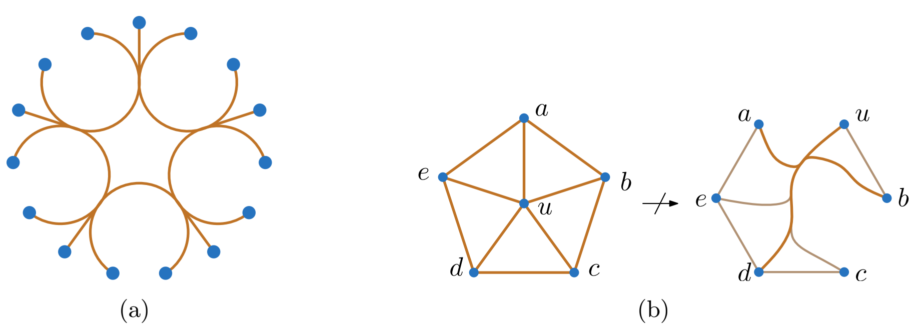


Figure 3: (a) Outerplanar strict confluent drawing of the GD2011 contest graph. (b) A graph with no outerplanar strict confluent drawing.

special cases of outerplanar strict confluent drawings as being the chordal bipartite graphs and distance-hereditary graphs respectively, so these graphs as well as the outerplanar graphs are all outerplanar strict confluent. The six-vertex wheel graph in Fig. 3(b) provides an example of a graph that does not have an outerplanar strict confluent drawing. (The central vertex u needs to be placed between two of the outer vertices, say, a and b. The smooth path from u to the opposite vertex d separates a and b, so there must be a junction shared by the u–d and a–b paths, creating a wrong adjacency with d.)

2 Preliminaries

In this section we first show some basic properties of strict confluent drawings and then provide linear upper bounds on their combinatorial complexity.

2.1 Basic Properties of Strict Confluent Drawings

We call an edge e in a drawing D *direct* if it consists only of a single arc (that does not pass through junctions). We call the angle between two consecutive arcs at a junction or vertex *sharp* if the two arcs do not form a smooth path; each junction has exactly two angles that are not sharp, and every angle at a vertex is sharp (so the number of sharp angles equals the degree of the vertex).

Let j be a junction. Then the two non-sharp angles at j separate the incident arcs into two disjoint sets A and B. Let a be an arc in A and $|B| > 1$. Then we say that j is a *split junction* for a since the path entering j via a splits at j into two or more paths. Conversely, if $|A| > 1$, we say that j is a *merge junction* for a since at least one more path merges with a at j.

Lemma 1. *Let $G = (V, E)$ be a graph, and let $E' \subseteq E$ be the edges of E that are incident to at least one vertex of degree 2. If G has a strict confluent drawing D, then it also has a strict confluent drawing D' in which all edges in E' are direct.*

Proof. Let v be a degree-2 vertex in G with two incident edges e and f. We consider the representation of e and f in D and modify D so that e and f are single arcs. There are two cases. If e and f leave v on two disjoint paths, then these paths have only merge junctions from v's perspective. We can simply separate these junctions from e and f as shown in Fig. 4(a). If, on the other hand, e and f share the same path leaving v, then their paths split at some point. We need to reroute the merge junctions prior to the split and separate the merge junctions after the split as shown in Fig. 4(b). This is always possible since v has no other incident edges. Because D was strict and these changes do not affect strictness, D' is still a strict confluent drawing and edges e and f are direct. □

Lemma 2. *Let G be a graph. If G has no $K_{2,2}$ as a subgraph, whose vertices all have degrees ≥ 3 in G, then G has a strict confluent drawing if and only if G is planar.*

Figure 4: The two cases of creating single arcs for edges incident to a degree-2 vertex.

Proof. Since every planar drawing is also a strict confluent drawing, that implication is obvious. So let D be a strict confluent drawing for a graph G without a $K_{2,2}$ subgraph, whose vertices all have degrees ≥ 3 in G. Since *large* junctions, where more than three arcs meet, can easily be transformed into an equivalent sequence of binary junctions, we can assume that every junction in D is binary, i.e., two arcs merge into one (or, from a different perspective, one arc splits into two). By Lemma 1 we can further transform D so that all edges incident to degree-2 vertices are direct. Now assume that there is a vertex u that has a path to a vertex v in D along which it visits a merge junction before visiting a split junction. Since D is strict, this situation implies the existence of a $K_{2,2}$ subgraph. All four vertices of that $K_{2,2}$ subgraph must have degree at least 3 as we have transformed all edges incident to degree-2 vertices into direct edges that do not pass through any junctions. This is a contradiction to the fact that G has no such subgraph. So the sequence of junctions on any u-v path in D consists of a number of split junctions followed by a number of merge junctions. But any such path can be unbundled from its junctions to the left and right and turned into a direct edge without creating arc intersections as illustrated in Fig. 5. Repeating the argument shows that D can be transformed into a standard planar drawing of G. □

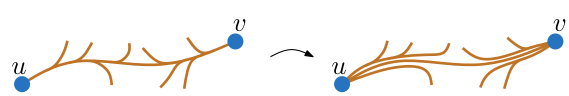


Figure 5: Any strict confluent drawing of a graph without a $K_{2,2}$ subgraph can be transformed into a standard planar drawing.

2.2 Combinatorial Complexity of Strict Confluent Drawings

Strict Confluent Drawings with Binary Junctions. We first assume that every vertex has degree one and every junction has degree three in the confluent drawing. Note that it is always possible to expand a confluent drawing to satisfy these restrictions. This expansion maintains the number of faces, and can only increase the number of arcs and junctions. Afterwards we will also consider junctions and vertices of higher degrees. We use the following lemma to reduce the number of cases we need to consider.

Lemma 3. *Let D be a strict confluent drawing. D cannot contain any smooth closed curves.*

Proof. Such a curve would either be irrelevant or would allow multiple paths (looping around the curve multiple times) for some pair of vertices. □

The following simple lemma is crucial for our bounds.

Lemma 4. *Every face in a strict confluent drawing D must have at least three sharp angles.*

Proof. Any face f in D is bounded by a sequence of arcs linking vertices and junctions. Recall that every angle at a vertex is sharp. If there is a junction with a sharp angle then we must also eventually reach a vertex if we trace any path through that junction since D has no loops by Lemma 3. Hence we can associate a vertex to each sharp angle of f. If there is no sharp angle in f, we have a smooth loop which contradicts Lemma 3. If there is a single sharp angle, we have a self loop for the associated vertex. If there are exactly two sharp angles, then the two associated vertices are connected by two different smooth paths along f. Both situations contradict the strictness of D. Hence f must have at least three sharp angles. □

In the following, n is the number of vertices, k is the number of junctions, m is the number of arcs, F is the number of faces, and c is the number of sharp angles.

Lemma 5. *Every strict confluent drawing has at most $2n-4$ faces, $5n-12$ junctions, and $8n-18$ arcs.*

Proof. By double counting we get that $2m = n + 3k$. The total number of sharp angles is $c = n + k$, and Lemma 4 implies that $c \geq 3F$, so that $n + k \geq 3F$. By combining the above relations with Euler's formula $n + k - m + F = 2$, we directly obtain that $F \leq 2n - 4$, $k \leq 5n - 12$, and $m \leq 8n - 18$. □

We can obtain similar bounds for outerplanar strict confluent drawings. Here we use F_{in} to denote the number of internal faces ($F_{in} = F - 1$).

Lemma 6. *Every outerplanar strict confluent drawing has at most $n-2$ internal faces, $3n-6$ junctions, and $5n-9$ arcs.*

Proof. By double counting we get that $2m = n + 3k$. The total number of sharp angles in internal faces is $c = k$ (every vertex is on the outer face), and Lemma 4 implies that $c \geq 3F_{in}$, so that $k \geq 3F_{in}$. By combining the above relations with Euler's formula $n+k-m+F_{in} = 1$, we directly obtain that $F_{in} \leq n - 2$, $k \leq 3n - 6$, and $m \leq 5n - 9$. □

Strict Confluent Drawings with Complex Junctions. If we allow junctions and vertices to have higher degree, then we can consider *minimal drawings*, that is, confluent drawings with as few arcs and junctions as possible. For strict confluent drawings that are minimal we can obtain stronger upper bounds on the combinatorial complexity. Consider a junction of degree three. We call the arc opposite of the sharp angle the *free arc*. Note that, regardless of what is on the other side of a free arc, we can always contract a free arc without changing the underlying graph represented by the confluent drawing. Furthermore, contracting a free arc will not influence other free arcs. That means that we can contract all free arcs simultaneously. Now note that in a minimal confluent drawing, all incident arcs of a vertex have been merged. Consider the unique arc incident to a vertex. If this arc is not free, then

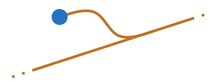


Figure 6: A junction with a pointless sharp angle.

the sharp angle of the adjacent junction must be bounded by this arc. In this case we call the respective sharp angle *pointless*; refer to Figure 6. This angle lies in the face that contains the vertex, but does not act as a "corner" of that face, and hence does not count towards the at least three sharp angles necessary for a face in a strict confluent drawing, as required by Lemma 4.

Lemma 7. *Every strict confluent drawing that is minimal has at most* $2n-6$ *junctions and* $5n-12$ *arcs. Every outerplanar strict confluent drawing that is minimal has at most* $n-3$ *junctions and* $3n-6$ *arcs.*

Proof. Let k' be the total number of pointless sharp angles and let n, k, and m be the number of vertices, binary junctions, and arcs, respectively, before performing any contractions. From Lemma 4 we obtain $n+k-k' \geq 3F$. Using Euler's formula $n+k-m+F=2$ as in the proof of Lemma 5 yields bounds $k \leq 5n-12-2k'$ and $m \leq 8n-18-3k'$ for binary junctions and arcs, respectively. Note that at least $n-2k'$ vertices must be adjacent to a free arc. Since every arc can be shared by at most two junctions/vertices, there must be at least $(k+n-2k')/2$ free arcs. Hence after contracting all free arcs there can be at most $k-(k+n-2k')/2 \leq 2n-6$ junctions and at most $m-(k+n-2k')/2 \leq 5n-12$ arcs (using $2m = n+3k$) in a minimal strict confluent drawing.

For outerplanar strict confluent drawings, note that all vertices and pointless sharp angles lie in the outer face. So by applying Lemma 4 to the inner faces we get $k-k' \geq 3F_{in}$. Using Euler's formula $n+k-m+F_{in}=1$ as in the proof of Lemma 6 yields bounds $k \leq 3n-6-2k'$ and $m \leq 5n-9-3k'$ for binary junctions and arcs, respectively. As before, after contracting all free arcs there can be at most $k-(k+n-2k')/2 \leq n-3$ junctions and at most $m-(k+n-2k')/2 \leq 3n-6$ arcs in a minimal outerplanar strict confluent drawing. □

Although we cannot prove a tight bound for (general) strict confluent drawings, the above bound for outerplanar strict confluent drawings is in fact tight.

Lemma 8. *Every outerplanar strict confluent drawing of a clique on* $n \geq 3$ *vertices has* $n-2$ *internal faces, and at least* $n-3$ *junctions and* $3n-6$ *arcs.*

Proof. To simplify the argument, we also refer to vertices as junctions. It is easy to see that every outerplanar strict confluent drawing of a graph with a triangle must contain an internal face f with three junctions. Also, since the underlying graph is a clique, the drawing cannot contain outside connections to an arc of f, as the origin of this connection will not be able to reach the junction of f opposite of the arc. Hence, every vertex must reach f through exactly one of the three junctions (exactly, since the drawing is strict). This means that we can partition the drawing (without f) into three parts: each part corresponds to one of the junctions of f and consists of all vertices that reach f through this junction plus the junction itself as a vertex. Additionally, each part of the drawing must again represent a clique on the respective vertices (including the junction of f). With this in mind we can setup the following recurrence relation with parameters x and y.

$$\begin{aligned} T(2) &= x \\ T(n) &= y + T(n_1 + 1) + T(n_2 + 1) + T(n_3 + 1) \\ &\qquad \text{where } n = n_1 + n_2 + n_3 \text{ and } n_1, n_2, n_3 \geq 1 \end{aligned}$$

This recurrence solves to $T(n) = x(2n - 3) + y(n - 2)$. Note that every arc of the face f must be present in the drawing and cannot be contracted. Every recursive step adds one internal face ($y = 1$) and a drawing of a 2-clique has no internal faces ($x = 0$), so we can use $(x, y) = (0, 1)$ to obtain $n - 2$ internal faces. Similarly, every recursive step adds three arcs ($y = 3$) and a drawing of a 3-clique must have three arcs ($T(3) = 3$, and hence $x = 0$), so we can use $(x, y) = (0, 3)$ to obtain $3n - 6$ arcs. For the number of junctions (including vertices), note that every junction is part of a subproblem ($y = 0$) and a drawing of a 3-clique must have three junctions ($T(3) = 3$, and hence $x = 1$), so we can use $(x, y) = (1, 0)$ to obtain $2n - 3$ junctions (including vertices), which implies $n - 3$ junctions excluding vertices. For the last two bounds it is important that every subproblem on two elements involves at least one element that is not a vertex (so the arc can be contracted), which is true if we start with $n \geq 3$ vertices. □

We summarize our discussion in the following theorem, which follows directly from Lemmas 5 and 8.

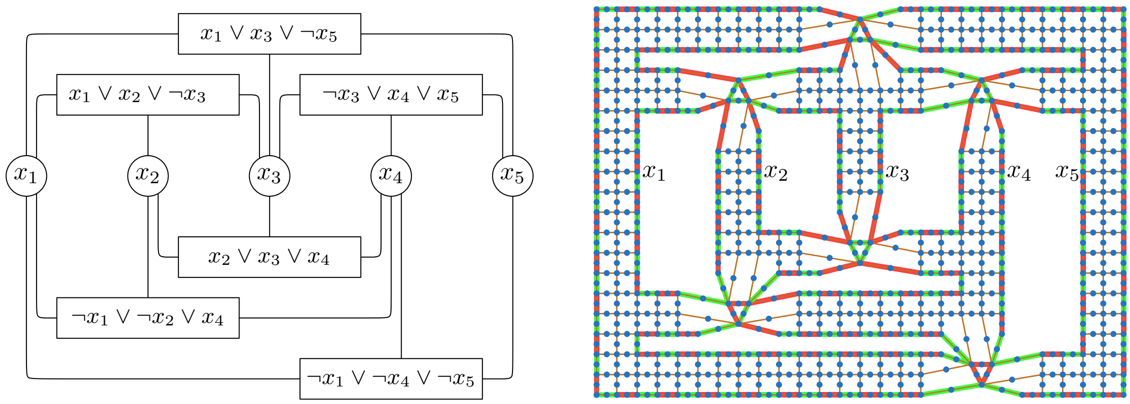
Theorem 1. *The combinatorial complexity of any strict confluent drawing D of a graph G, i.e., the total number of arcs, junctions, and faces in D, is linear in the number of vertices of G.*

Theorem 1 is in contrast to previous methods for confluent drawings of interval graphs [4] and for drawing confluent Hasse diagrams [9], both of which may produce (non-strict) drawings with quadratically many features.

3 Computational Complexity

We will show by a reduction from planar 3-SAT [15] that it is NP-complete to decide whether a graph G has a strict confluent drawing in which all edges incident to degree-2 vertices are direct. By Lemma 1, this is enough to show that it is also NP-complete to decide if G has any strict confluent drawing.

Consider the subdivided grid graph (a grid with one extra vertex on each edge). In this graph, all edges are adjacent to a degree 2 vertex. Since a grid graph more than one square wide has only one fixed planar embedding (up to choice of the outer face), the subdivided grid graph has only one confluent embedding in which all edges are direct. We will base our construction on a number of such grids.



(a) (b)

Figure 7: (a) A planar 3-SAT formula. (b) The corresponding global frame of the construction: one grid graph per variable, with some vertices identified at each clause. Green boundary edges correspond to positive literals, red edges to negated literals. For easier readability the grids in this figure are larger than strictly necessary.

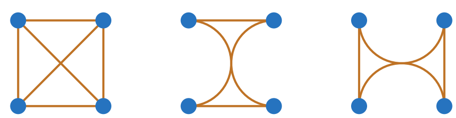


Figure 8: K_4 and its two strict confluent drawings, without moving the vertices and keeping all arcs inside the convex hull of the vertices.

Let S be a planar 3-SAT formula. Globally speaking, we will create a grid graph for each variable of S, of size depending on the number of clauses that the variable appears in. The external edges of this grid graph are alternatingly colored green and red. We connect the variable graphs by identifying certain vertices: for each of the three variables that appear in a clause, we select one subdivided edge (that is, three vertices connected by two edges) on the outer face, and identify the endpoints of these edges into a triangle of subdivided edges (that is, a 6-cycle). We choose a green edge for a positive occurrence of the variable and a red edge for a negated occurrence; the purpose of this will become clear below. We call the resulting graph F the *frame* of the construction; all edges of F are adjacent to a degree-2 vertex and F has only one planar embedding (up to choice of the outer face). Figure 7 shows an example.

The main idea of the construction is based on the fact that K_4, when drawn with all four vertices on the outer face, has exactly two strict confluent drawings: we need to create a junction that merges the diagonal edges with one pair of opposite edges, and we can choose the pair. Figure 8 illustrates this. We will add a copy of K_4 to every cell of the frame graph F. Recall that every cell, except for the triangular clause faces, is a subdivided square (that is, an 8-cycle). We add K_4 on the four grid vertices (not the subdivision vertices). The edges that connect external grid vertices are called *literal edges*. Figure 9(a) shows this for a small grid. Since neighboring grid cells share a (subdivided) edge, the K_4's are not

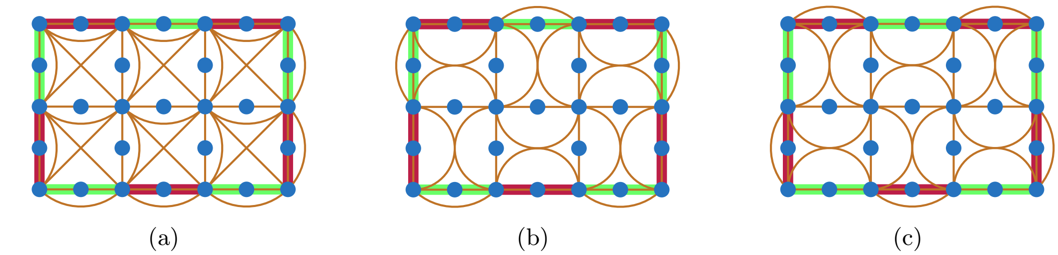


Figure 9: (a) A variable gadget consists of a grid of K_4's. Green (light) edges of the frame highlight normal literals, red (dark) edges negated ones. (b) One of the two possible strict confluent drawings, corresponding to the value *true*. (c) The other strict confluent drawing, corresponding to *false*.

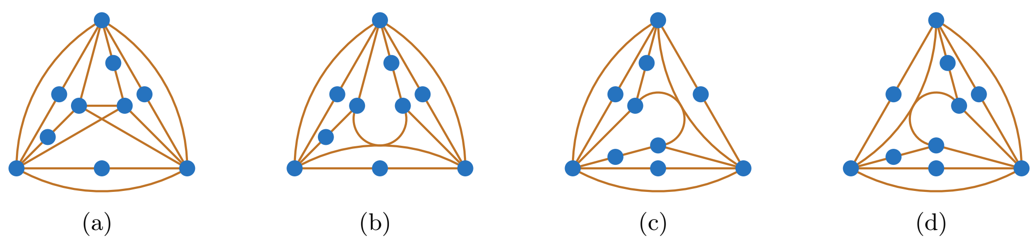


Figure 10: (a) The input graph of the clause. (b, c, d) Three different strict confluent drawings.

edge-independent. This implies that in a strict confluent drawing, we cannot "use" such a common edge in both cells. Therefore, we need to orient the K_4-junctions alternatingly, as illustrated in Figures 9(b) and 9(c). If the grid is sufficiently large (every cell is part of a larger at least size-(2×2) grid) these choices are completely propagated through the entire grid, so there are two structurally different possible embeddings, which we use to represent the values *true* and *false* of the corresponding variable. For every green edge of the frame in the *true* state and every red edge in the *false* state there is one remaining literal edge in the outer face, which can still be drawn either inside or outside their grid cells. In the opposite states these literal edges are needed inside the grid cells to create the K_4 junctions. The availability of at least one literal edge (corresponding to a *true* literal) is important for satisfying the clause gadgets, which we describe next.

Inside each triangular clause face, we add the graph depicted in Figure 10(a). This graph has several strict confluent drawings; however, in every drawing at least one of the three outer edges needs to be drawn inside the subdivided triangle.

Lemma 9. *There is no strict confluent drawing of the clause graph in which all three triangle edges are drawn outside. Moreover, there is a strict confluent drawing of the clause graph with two of these edges outside, for every pair.*

Proof. Recall that by Lemma 1 the subdivided triangle must be embedded as a 6-cycle of direct arcs. To prove the first part of the lemma, assume that the triangle edges are all drawn outside this cycle. The remainder of the graph has no 4-cycles without subdivision vertices (that is, no $K_{2,2}$ with higher-degree vertices), so by Lemma 2 it can only have a

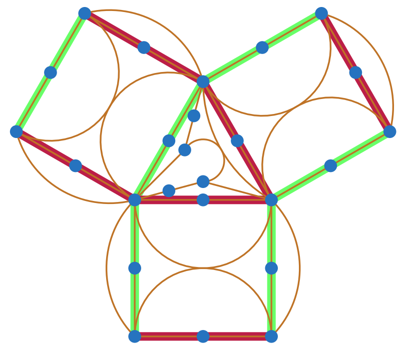


Figure 11: Three variables attached to a clause gadget. The top left variable occurs in the clause as a positive literal, the others as negative literals. The clause can be satisfied because the top right variable is set to *false*.

strict confluent drawing if it is planar. However, it is a subdivided K_5, which is not planar. To prove the second part of the lemma, we refer to Figures 10(b), 10(c) and 10(d). □

We described the reduction from a planar 3-SAT instance to a graph consisting of variable and clause gadgets. Next we show that this graph has a strict confluent drawing if and only if the planar 3-SAT formula is satisfiable. For a given satisfying assignment we choose the corresponding embeddings of all variable gadgets. The assignment has at least one *true* literal per clause, and correspondingly in each clause gadget one of the three literal edges can be drawn inside the clause triangle, allowing a strict confluent drawing by Lemma 9. Conversely, in any strict confluent drawing, each clause must be drawn with at least one literal edge inside the clause triangle by Lemma 9, so translating the state of each variable gadget into its truth value yields a satisfying assignment.

To show that testing strict confluence is in NP, recall that by Theorem 1 the combinatorial complexity of the drawing is linear in the number of vertices. Thus the existence of a drawing can be verified by guessing its combinatorial structure and verifying that it is planar and a drawing of the correct graph.

Theorem 2. *Deciding whether a graph has a strict confluent drawing is* NP*-complete.*

4 Canonical Diagrams for Outerplanar Strict Confluent Drawings

We now restrict our attention to outerplanar drawings. In this section, we introduce the notion of a *canonical diagram* of G. The canonical diagram of G serves as a unique representation of all outerplanar strict confluent drawings of G (with the vertices in a given order on the outer face). We first show that the canonical diagram exists if and only if an outerplanar strict confluent drawing exists. Next, we show that the canonical diagram is indeed unique. The uniqueness of the canonical diagram will be crucial for the algorithm in Section 5.

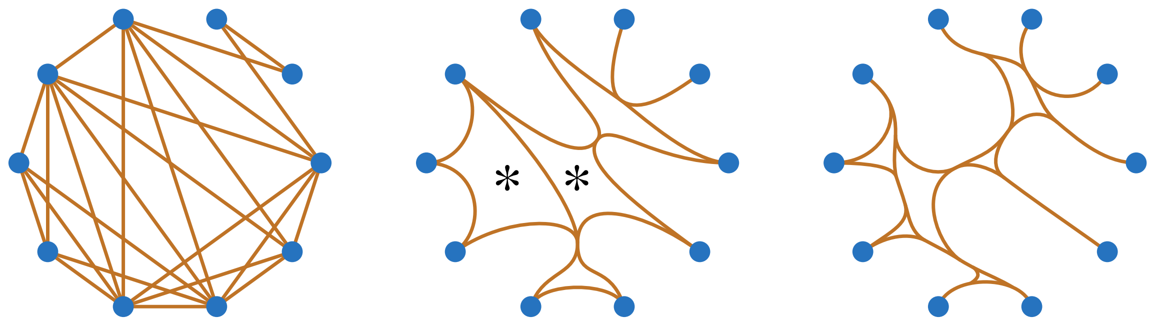


Figure 12: Three views of the same graph as a node-link diagram (left), canonical diagram (center), and outerplanar strict confluent drawing (right).

4.1 Canonical Diagrams

We define a *canonical diagram* to be a collection of junctions and arcs connecting the vertices in the given order on the outer face (as in a confluent drawing), but with some of the faces of the diagram *marked*, satisfying additional constraints enumerated below. Figure 12 shows a canonical diagram and an outerplanar strict confluent drawing of the same graph. In such a diagram, a *trail* is a smooth curve from one vertex to another that follows the arcs (as in a confluent drawing) but is allowed to cross the interior of marked faces from one of its sharp corners to another. The constraints of a canonical diagram are:

1. Every arc is part of at least one trail.
2. Any two trails between the same two vertices must follow the same sequence of arcs and faces.
3. Each marked face must have at least four angles, and all its angles must be sharp.
4. Each arc must have either sharp angles or vertices at both of its ends.
5. For each junction j with exactly two arcs in each direction, let f and f' be the two faces with sharp angles at j. Then it is not allowed for f and f' to both be either marked or a triangle (a face with three angles, all sharp).

A canonical diagram represents a graph G in which the edges in G correspond to trails in the diagram.

4.2 Equivalence of Canonical Diagrams and Outerplanar Strict Confluent Drawings

In the following we show that a graph G has a canonical diagram if and only if it has an outerplanar strict confluent drawing. Let j be a junction of a canonical diagram D. Then define the *funnel* of j to be the 4-tuple of vertices a, b, c, d where a is the vertex reached by a path that leaves j in one direction and continues as far clockwise as possible, b is the most counterclockwise vertex reachable in the same direction from j, c is the most clockwise vertex reachable in the other direction, and d is the most counterclockwise vertex reachable in the other direction. Note that none of the paths from j to a, b, c, and d can intersect each other without contradicting the uniqueness of trails. We call the circular intervals of vertices $[a, b]$ and $[c, d]$ (in the counterclockwise direction) the *funnel intervals* of the respective funnel. We say a circular interval $[a, b]$ is *separated* if either a and b are not adjacent in G, or there exists a junction in the canonical diagram with funnel intervals $[a, e]$ and $[f, b]$, where $e, f \in [a, b]$.

Lemma 10. *In every outerplanar strict confluent drawing or canonical diagram, in which each vertex has at least one incident edge, there must be a pair of vertices that are consecutive on the outer face of the drawing and adjacent in the corresponding graph. If there are at least three vertices, then there must be at least two such pairs.*

Proof. Let uv be an adjacent pair of vertices that are as close as possible to each other on the outer face, as measured by the smaller of the two sequences of vertices from u to v and from v to u around the outer face. Then uv must be consecutive. For, if they were not consecutive, then there would be a vertex w between them. Any trail from w to one of its neighbors would have to cross the trail for uv, causing w to be adjacent to one of u or v. But this would contradict the choice of uv as being as close as possible for an adjacent pair.

Next, suppose that there are three or more vertices and let $u'v'$ be an adjacent pair of vertices that are as close as possible to each other by a different distance, the size of the sequence of vertices from u' to v' or v' to u' around the outer face, whichever of these two sequences does not contain the consecutive pair uv. Note that $u'v'$ cannot equal uv, because uv is the pair with the largest distance by this measure. Again, we claim that $u'v'$ must be consecutive, for if they were separated by another vertex w' then the trail from w' to one of its neighbors would have to cross the trail for $u'v'$, causing w' to be adjacent to one of u' or v' and leading to a contradiction. □

Using these observations, we can argue that in fact, there cannot be any cycles in the dual graph of the marked faces in a canonical diagram (refer to Figure 12).

Lemma 11. *In every canonical diagram the dual graph of the set of marked faces forms a forest.*

Proof. Suppose for a contradiction that this dual graph contains an induced cycle C. It is not possible for marked faces to entirely surround a single junction of the diagram, because they all have sharp angles at the junction and every junction has two non-sharp angles. Therefore, the part of the diagram inside C must consist of one or more faces bounded by arcs of the faces in C. These surrounded faces may form a single connected region R, or they may form multiple connected regions separated by junctions that appear more than once along the boundary of C; in the latter case, these regions and the junctions that connect them form a tree, and we may choose R to be a leaf of the tree. Thus, in either of these two cases there exists a region R consisting of a set of neighboring faces of the diagram, in which all but at most one of the junctions on the boundary of R span an angle of 2π (the one exceptional junction being the one that connects R to other connected regions within C).

The part of the diagram within C can itself be viewed as a confluent drawing, within which each of the junctions that spans an angle of 2π is connected to at least one other

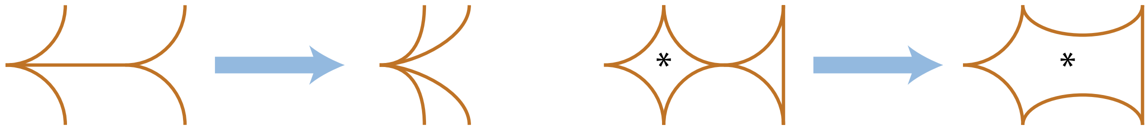


Figure 13: The simplification operations that transform an outerplanar strict confluent drawing into its canonical diagram: contraction of arcs without sharp angles at both ends (left) and merger of marked faces and triangles (right).

junction (otherwise the arcs into that junction could not be part of a complete trail). By Lemma 10, there exist two consecutive junctions on the boundary of C that are connected by a smooth path within R. This path, together with a smooth path connecting the same two junctions within the marked face of C that contains them both, forms a continuous smooth loop in the original diagram, contradicting Lemma 3. This contradiction shows that C cannot exist. □

Theorem 3. *A graph G may be represented by a canonical diagram if and only if it may be represented by an outerplanar strict confluent drawing.*

Proof. To form a canonical diagram from an outerplanar strict confluent drawing of G, repeatedly perform the following two simplifications (Figure 13):

- Contract any arc that has a sharp angle or a vertex at at most one of both ends.
- Let j be any junction with exactly two arcs in each direction, let f and f' be the two faces with sharp angles at j, and suppose that f and f' are both either marked or a triangle. Then merge the two faces f and f' by connecting them through j, removing the junction from the drawing, and mark the resulting merged face.

To show that applying these reduction rules exhaustively terminates with a canonical diagram, we observe that the defining properties of a canonical diagram hold. Both operations preserve that each arc is part of a trail, that each edge of G is represented by a combinatorially unique trail, that each non-adjacent pair in G has no trail, and that each marked face has the correct shape (at least four angles, all sharp). Moreover, the first operation removes all arcs that do not have sharp angles or vertices at both ends and the second operation merges all pairs of marked faces or triangles that are linked via a degree-4 junction. Each step reduces the number of arcs, so this simplification process eventually terminates with a canonical diagram.

Conversely, any canonical diagram can be converted into an equivalent outerplanar strict confluent drawing by repeatedly reversing the second of the two simplification operations. The reverse of this operation splits a marked face f by pinching together two nonadjacent sides to form a new junction with four incident arcs; the two faces formed from f are marked if they have more than three angles, or left unmarked if they are triangles. It is not possible to perform this pinching off step if one or both of the sides of the step is the boundary of another marked face. However, by Lemma 11 it is always possible to choose a marked face to simplify that forms an isolated vertex or leaf of the dual graph of the set of remaining marked faces; such a face can always safely be simplified. □

4.3 Uniqueness of Canonical Diagrams

Next, we show that if a canonical diagram exists then it is unique. In a canonical diagram D, for any subset $K \subseteq V$ of its vertices, define a *pseudotriangle* for K to be a face of D that has exactly three sharp junctions, each of which is part of a trail from the face to a vertex in K, and for which all other boundary junctions (if they exist) do not lead to vertices in K. A *side* of a pseudotriangle is the path of arcs connecting two of its sharp angles.

Lemma 12. *Let G be represented by a canonical diagram D and let K be a clique of two or more vertices in G. Then the arcs and faces traversed by trails connecting vertices of K form a tree of arcs, marked faces, and pseudotriangles for K, connected to each other at junctions. In this tree, each junction is incident to exactly two arcs, marked faces, or pseudotriangles, one in each direction.*

Proof. We use induction on the size of K; as a base case, the result is clearly true for $|K| = 2$, for which the single trail forms a path of arcs and marked faces. If $|K| > 2$ and v is an arbitrary vertex of K, then the trails through $K - v$ form a tree of arcs, marked faces, and pseudotriangles by the induction hypothesis. In order to reach every vertex of $K - v$, and avoid making multiple connections to any vertex, the trails from v to $K - v$ can only connect to this tree in one of three ways:

1. It may be incident to one of the sharp angles of a marked face of D that already belongs to the tree.
2. It may be incident to a sharp angle of a marked face of D that is not already in the tree, but that has an arc in the tree, causing the face to be added to the tree.
3. It may be incident to a sharp angle of a pseudotriangle of D that is not already in the tree, but whose opposite side is already in the tree, again causing the face to be added to the tree.

In any of these cases, outside of the tree for $K - v$, all the trails to v must follow the same sequence of arcs and marked faces, for to do otherwise would violate the uniqueness of trails in canonical diagrams. The tree for $K - v$, together with the arcs and marked faces traversed by the rest of the trail from this tree to v, forms another structure of the same type, and contains trails from all vertices of $K - v$ to v. □

Lemma 13. *For every junction j of a canonical diagram with funnel intervals $[a, b]$ and $[c, d]$, at least one of the intervals must be separated.*

Proof. For the sake of contradiction, assume that $[a, b]$ and $[c, d]$ are not separated. Then the vertices a, b, c, and d form a clique. By Lemma 12 the drawing would have two pseudotriangles or marked faces, one on each side of j. The boundaries of these faces adjacent to j can be the only arcs into j, for otherwise one of a, b, c, or d would not be in the funnel (see Fig. 14). Now assume there is a junction j' between a and b (or c and d) on the boundary of the face opposite to j (in this case the face must be a pseudotriangle). Consider the path that leaves j' in the direction of a and continues as far clockwise as possible. This path cannot pass

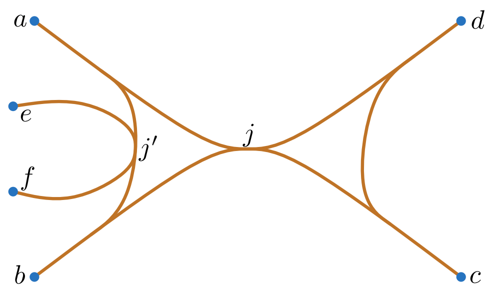


Figure 14: The configuration of Lemma 13.

through the pseudotriangle, because that would result in multiple paths between b and either c or a, which is not allowed. Hence this path must reach a, for otherwise a would not be a funnel vertex of j. Similarly, the path that leaves j' in the direction of b and continues as counterclockwise as possible must reach b. The other paths of the funnel must end at a vertex in the circular interval $[a, b]$. This would imply that $[a, b]$ is separated, which contradicts our assumption. Thus, the faces on either side of j must be marked faces or triangles. But this configuration violates Condition 5 that there be no junction j with four arcs in which the two faces having sharp angles at j are marked or triangles. □

Lemma 14. *Let D and D' be canonical diagrams for the same graph G with the same vertex ordering. Then for every junction j of D, there must be a junction j' in D' with the same funnel.*

Proof. For a junction j of D with funnel intervals $[a, b]$ and $[c, d]$, let $[a, b]$ be the separated funnel interval. We prove the lemma by induction on the cardinality of $[a, b]$. The existence of the junction implies that (a, c) and (b, d) are edges of G. Therefore, the trail in D' between a and c must cross the trail between b and d in the canonical diagram. This can happen in three ways (see Fig. 15): (1) the trails from a and b merge and then split towards c and d, (2) the trails from b and c merge and then split towards a and d, or (3) the trails cross inside a marked face F from which each vertex a, b, c, and d is reached by a different junction of F.

In case (1) the trails can travel together along a sequence p. But because $[a, b]$ and $[c, d]$ are the funnel intervals for j, there can be nothing entering or leaving p between the merge and split points, and p cannot contain a nonzero number of arcs without violating Condition 4 on arcs without sharp angles at both ends. Therefore, the merge point of the trails from a and b is also the split point of the trails to c and d, and forms a single junction j' in D' with the same funnel.

In the latter two cases, a and b are connected by a trail in D', therefore they are adjacent in G. Hence there must exist a junction ι in D with funnel intervals $[a, e]$ and $[f, b]$, where $e, f \in [a, b]$. If $[a, b]$ contains less than two vertices besides a and b, then this is not possible (base case). Otherwise, by induction, there must exist a corresponding junction ι' in D' with funnel intervals $[a, e]$ and $[f, b]$. This junction must lie along the trail between a and b. By the properties of a funnel, neither e nor f can be connected to c or d. This directly makes case (2) impossible, since such a connection is necessary in that configuration. In case

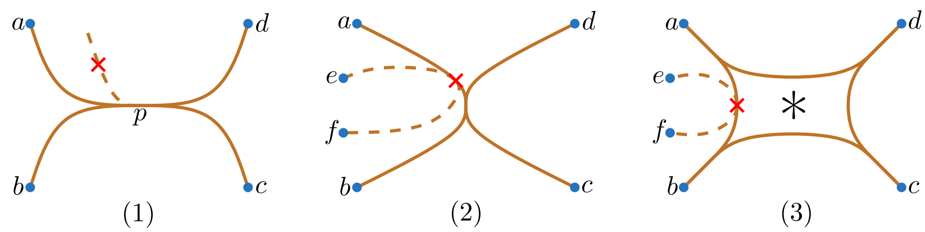


Figure 15: The cases of Lemma 14.

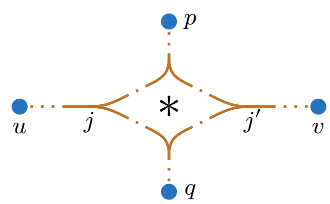


Figure 16: The configuration in Lemma 16.

(3) this implies that ι' must lie on the part of the trail between a and b that is along (or through) F. This is not possible if the trail passes through F, so ι' must be at the boundary of F. But this makes the marked face F invalid, as it may have only sharp angles. □

Lemma 15. *Let j be a junction of canonical diagram D with funnel a, b, c, d, and let uv be an edge of the graph G represented by D. Then the trail for edge uv passes through j if and only if exactly one of u and v is in the closed interval $[a, b]$ and exactly one of u and v is in the closed interval $[c, d]$.*

Proof. If the trail passed through j but u or v was outside these intervals, it would be reached from j by a more extreme path than one of a, b, c, or d, violating the definition of a funnel. If both are inside the same interval, but the trail between them goes through j, then the diagram violates the requirement that trails be unique. And if u and v are both inside different intervals, then the path must either go through j or it must cross the funnel twice, again causing a violation of the requirement that trails be unique. □

Lemma 16. *Let D be a canonical diagram in which two junctions j and j' are consecutive on the trail from u to v. Then this trail passes across a marked face from j to j' if and only if there exist vertices p and q, in the cyclic order u, p, v, q, such that u, v, p, and q form a clique, and such that p and q are both in the same interval as v in the funnel for j and in the same interval as u in the funnel for j'.*

Proof. The situation is illustrated in Figure 16. Suppose first that the trail crosses a marked face. This face must have sharp angles on both sides of trail uv; let p and q be any two vertices reachable from two angles on opposite sides. Then there also exist trails from u and v to p and q that diverge from the trail from u to v within the marked face. The containment of p and q within the stated intervals follows from Lemma 15.

In the other direction, suppose that the trail passes consecutively through j and j' and that there exist p and q satisfying the conditions of the lemma. Since u, v, p, and q form a clique, Lemma 12 implies that this clique is represented either by a marked face with at least four sharp angles connecting to all four of these vertices, or by two marked faces or triangles connected by a sequence of junctions, arcs, and additional marked faces. However, because of the assumption in the lemma that p and q belonging to certain intervals, all of these faces must lie between j and j'; the additional assumption that j and j' are consecutive on the trail means that there is only one possibility, a single marked face with separate sharp angles connecting to the four vertices u, v, p, and q. Thus, as the lemma states, the trail crosses a marked face. □

Lemma 17. *Let D and D' be canonical diagrams for the same graph G with the same vertex ordering. Then for every arc from junction j to j' of D, there must be an arc in D' connecting the corresponding junctions.*

Proof. Let uv be an edge of G whose trail in D uses the arc. The trails in D and D' from u to v both go through the same sets of junctions by Lemma 15. On the trail in D from u to v, the junctions are monotonic in their ordering by the size of the sets of reachable vertices in each direction, and the same is true in D', so the trails go through the same junctions in the same ordering. Therefore, in D', the trail from u to v also goes from j to j' with no intervening junctions. The only way it can avoid using an arc that satisfies the lemma is for it to cross a marked face instead of an arc, but this would violate Lemma 16 for either D or D'. □

Theorem 4. *Every two canonical diagrams for the same graph G with the same vertex ordering are isomorphic.*

Proof. This follows from Lemma 14 (showing that they have the same set of junctions) and Lemma 17 (showing that they have the same set of arcs). □

5 Algorithms for Computing Outerplanar Strict Confluent Drawings

By using the properties of canonical diagrams derived in the previous section, we may obtain an algorithm that constructs a canonical diagram and an outerplanar strict confluent drawing of a given cyclically-ordered graph G, or reports that no drawing exists, in time and space $\Theta(n^2)$ (Section 5.1). We further show in Section 5.2 that the drawing can be constructed such that every arc consists of at most two circular arcs.

5.1 Algorithm for Constructing Canonical Diagrams

Steps 1–3 of the algorithm, detailed below, build some simple data structures that speed up the subsequent computations. Step 4 discovers all of the funnels in the input, from which it constructs a list of all of the junctions of the canonical diagram. Step 5 connects these junctions into a planar drawing, a subset of the canonical diagram. Step 6 builds a graph for each face of this drawing that will be used to complete it into the entire canonical diagram, and step 7 uses these graphs to find the remaining arcs of the diagram and to determine which faces of the diagram are marked. Step 8 checks that the diagram constructed by the previous steps correctly represents the input graph, and step 9 splits the marked faces, converting the diagram into a strict confluent drawing.

1. Number the vertices clockwise around the boundary cycle from 0 to $n-1$.

2. Build a table $T(i,j)$, containing for each pair i,j, the number of ordered pairs (i',j') with $i' \le i$, $j' \le j$, and vertices i' and j' adjacent in G. This is simply a summed area table for the adjacency matrix of G which can be computed in $O(n^2)$ time. By performing a constant number of lookups in this table we may determine in constant time how many edges exist between any two disjoint intervals of the boundary cycle.

3. Build a table $N^{\circlearrowright}(u,v)$ that lists, for each ordered pair u,v of vertices, the neighbor w of u that is closest in clockwise order to v. That is, w is adjacent to u, and the interval from v clockwise to w contains no other neighbors of u. The table entries of the form $N^{\circlearrowright}(u,\cdot)$ can be found in linear time by a single counterclockwise scan. Repeat the same construction in the opposite orientation to build a table $N^{\circlearrowleft}(u,v)$.

4. Conceptually, for each separated interval $[a,b]$, let c be the next neighbor of a that is counterclockwise of b, and let d be the next neighbor of b that is clockwise of a. If (i) c is a neighbor of b, (ii) d is a neighbor of a, (iii) a is the next neighbor of c that is counterclockwise of d, and (iv) b is the next neighbor of d that is clockwise of c, then (if a confluent diagram exists) a,b,c,d must form the funnel of a junction, and all funnels have this form.

   To compute all funnels, we go through all circular intervals in increasing order of their cardinalities. For every separated interval, we check the conditions (i)-(iv) described above. These conditions can be checked in $O(1)$ time using the adjacency matrix and the tables $N^{\circlearrowright}$ and $N^{\circlearrowleft}$. For each discovered funnel, we mark the intervals that are separated by the corresponding junction. This way we can also check in $O(1)$ time whether a circular interval is separated. If the number of funnels exceeds the linear bound of Lemma 6 on the number of junctions in a confluent drawing, abort the algorithm.

5. Create a junction for each of the funnels found in step 4. For each vertex v, make a set J_v of the junctions whose funnel includes that vertex; if they are to be drawn as part of a canonical diagram, the junctions of J_v need to be connected to v by a confluent tree. For any two junctions $j, j' \in J_v$, it is possible to determine in constant time whether j is an ancestor of j' in this tree: if j is an ancestor of j', then the funnel interval of j' (the one not containing v) is a subset of the funnel interval of j. If j is not an ancestor of j' or vice versa, then the funnel intervals of j and j' are disjoint, and it can easily be checked if one is clockwise of the other. Construct the trees of junctions and their planar embedding in this way. The result of this stage of the algorithm should be a planar embedding of part of the canonical diagram consisting of all vertices and junctions, and the subset of the arcs that are part of a path from a junction to one of its funnel vertices. Check that the embedding is planar by computing its Euler characteristic, and abort the algorithm if it is not.

6. For each face f of the drawing created in step 5, and each pair j, j' of junctions belonging to f, use table T to test whether there is an edge whose trail passes through both j and j'. An edge like this must have one endpoint in the funnel interval of j (opposite of f) and the other endpoint in the funnel interval of j' (opposite of f). This results in a graph H_f in which the vertices represent the vertices or junctions on the boundary of f and the edges represent pairs of vertices or junctions that must be connected, either by an arc or by shared membership in a marked face. The remaining arcs to be drawn in f will be exactly the edges of H_f that are not crossed by other edges of H_f; the marked faces in f will be exactly the faces (after these additional arcs are drawn) that contain pairs of crossing edges of H_f.

7. Within each face f of the drawing so far, build a table using the same construction as in step 2 that can be used to determine the existence of a crossing edge for an edge in H_f in constant time. Use this data structure to identify the crossed edges, and draw an arc in f for each uncrossed edge. For each face g of the resulting subdivision of f, if g has four or more vertices or junctions, find two pairs that would cross and test whether both pairs correspond to edges in H_f; if so, mark g.

8. Construct a directed graph that has a vertex for each vertex of G, two vertices for each junction of the diagram (one in each direction), two directed edges for each arc, and a directed edge for each ordered pair of sharp angles that are non-consecutive in a marked face. By performing a depth-first search in this graph, determine whether there exist multiple smooth paths in the resulting drawing from any vertex v of G to any other point in the drawing, and abort the algorithm if any such pair of paths is found. Determine the set of vertices of G reachable from v and verify that it is the same set of vertices that are reachable in the original graph. Additionally, verify that the diagram satisfies the requirements in the definition of a canonical diagram. Abort the algorithm if any inconsistency is found in this step.

9. Convert the canonical diagram into its equivalent outerplanar strict confluent drawing by repeatedly pinching together two non-adjacent sides of each marked face and thus splitting it into smaller marked faces or triangles as described in Theorem 3. This requires to compute and maintain the dual graph of the canonical diagram in order to pick a suitable marked face in each step. Due to the linear complexity of the resulting drawing shown in Theorem 1 at most $O(n)$ pinching steps are needed; for the same reason computing the dual graph takes $O(n)$ time.

The next theorem summarizes the properties of our algorithm.

Theorem 5. *For a given n-vertex graph G, and a given circular ordering of its vertices, it is possible to determine whether G has an outerplanar strict confluent drawing with the given vertex ordering, and if so to construct one, in time $O(n^2)$.*

Proof. Given the fact that the resulting drawing can have only $O(n)$ complexity, it is easy to verify that the above algorithm runs in $O(n^2)$ time. For the correctness, note that we explicitly check the validity of the canonical diagram in step 8 of the algorithm. Thus, using

Theorem 3, if G does not have an outerplanar strict confluent drawing with the given vertex ordering, then this will be detected by the algorithm. It remains to be shown that, if the algorithm does not return an outerplanar strict confluent drawing, then such a drawing does not exist for G with the given vertex ordering. In step 4 the algorithm terminates if the number of funnels exceeds $3n-6$. This is correct, since by Lemma 6 every outerplanar strict confluent drawing can have at most $3n-6$ junctions, and every funnel has an associated junction. In step 5 the algorithm terminates if the embedding is not planar. The fact that this can be checked using the Euler characteristic is folklore (see e.g. [17]). Finally, in step 8 the algorithm terminates if any inconsistency was found in the resulting canonical diagram. Since we only add junctions and arcs that are necessary, and we only mark faces when we need to, the resulting canonical diagram can only be incorrect if none exists for G with the given vertex ordering. Thus, the algorithm described above is correct. □

5.2 Drawings with low curve complexity

Suppose that we are given a topological description of an outerplanar strict confluent drawing D of a connected graph G, describing the tangency pattern and ordering of the arcs at each junction. It still remains to draw D (or possibly an equivalent but combinatorially different outerplanar strict confluent drawing) in the plane using concrete curves for its arcs. If we ignore the tangency requirements at its junctions, the arcs and junctions of D form a planar graph, but applying standard planar graph drawing methods will generate arcs that may not be smooth and that are not tangent to each other at the junctions. So how are we to draw D? In this section we use a circle packing method to draw D with a small number of circular arcs for each arc of D. Thus, these drawings have low *curve complexity* in the sense of Bekos et al. [1], but with this complexity measured along arcs of the confluent diagram rather than edges of another type of graph drawing.

Given such a drawing D, let D' be a modified version of D in which every junction is incident to exactly three arcs, formed from D by suppressing two-arc junctions and splitting junctions with more than three arcs. Assume also (again by adding more junctions if necessary) that each vertex in D' has only a single arc incident to it.

Given the topological diagram D', we form a planar graph H that has a vertex for each vertex or junction of D', and an edge for each arc of D'. Additionally, we create an edge in H for each pair of vertices that are consecutive in the cyclic ordering of the vertices around the disk containing the drawing.

Lemma 18. *H is planar, 3-regular, and 3-vertex-connected.*

Proof. By construction, the graph H is planar and 3-regular. It remains to show that it is 3-vertex-connected. We first argue about the vertices of G (assuming that G is a connected graph). Any pair of vertices of G is connected by at least one (not necessarily smooth) path through D' and two disjoint paths around the outer face. Thus it is clear that all vertices of G belong to the same 3-connected component C of H. For the sake of contradiction, assume that C does not contain the entire graph H. Hence there exists some cut vertex or separation pair in H. If there is a cut vertex, then the component not containing C either

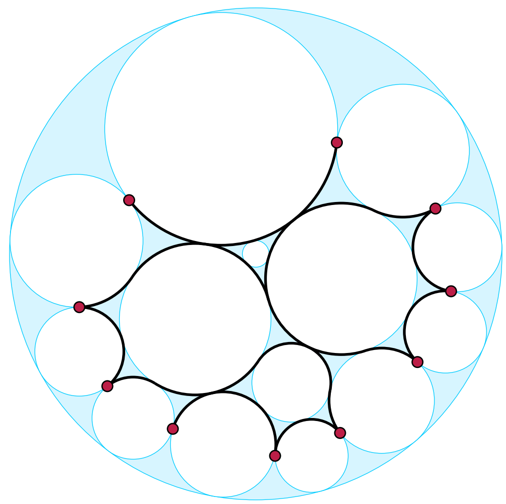


Figure 17: Construction of an outerplanar strict confluent diagram from a circle packing. The vertices of the diagram correspond to triangular gaps adjacent to the outer circle, and the junctions of the diagram correspond to the remaining triangular gaps.

has edges that do not belong to any smooth path or there must be a self-loop, both of which are forbidden in a strict confluent drawing. If there is a separation pair, we argue that the component not containing C has at least one junction vertex j, and it has to be of degree 3 since H is 3-regular. Since all edges in a strict confluent drawing belong to a smooth path between two vertices in C, the three subpaths connecting to j must either form a self-loop or two different smooth paths between the same sets of vertices as they are all required to run through the separation pair. This is again forbidden in a strict confluent drawing. □

Theorem 6. *Let D be an outerplanar strict confluent drawing of a graph G, given topologically but not geometrically. Then we can construct an outerplanar strict confluent drawing of G in which each arc of the drawing is represented by a smooth curve that is either a circular arc or the union of two circular arcs.*

Proof. By the Koebe–Thurston–Andreev circle packing theorem (which we can apply due to the properties of H shown in Lemma 18), there exists a system C of circles representing the faces of H, such that two circles are adjacent exactly when the corresponding faces share an edge. We may assume (by performing a Möbius transformation if necessary) that the outer circle of this circle packing corresponds to the outer face of H. C may be found efficiently (although not in strongly polynomial time) by a numerical iteration that quickly converges to the system of radii of the circles, from which their centers can also be computed easily [2,18].

Each vertex of G corresponds in C to one of the triangular gaps between the outer circle and two other circles, and may be placed at the point of tangency of the two non-outer circles (one of the vertices of this triangle); see Fig. 17. The junctions in D' lie at the meeting point of three faces of H, and correspond in C to the remaining triangular gaps between three circles. A confluent drawing of G may be formed by removing the outer circle, removing all circular arcs bounding the triangular gaps incident to the outer circle, and in each remaining triangular gap removing the arc that is on the other side of the sharp angle. The resulting drawing contracts some edges of D' to form junctions with four incident arcs, but this does not affect the correctness of the drawing. In the resulting drawing, arcs of the diagram that have merge points or vertices at both of their endpoints are drawn as two circular arcs (possibly both from the same circle); other arcs of the diagram are drawn as a single circular arc. □

We remark that the area of strict confluent drawings based on circle packings as described in Theorem 6 can in fact be exponential, e.g., for nested triangle graphs. However, it is also at most (singly) exponential since every planar graph has a circle packing, in which the ratio of the radii of any two neighboring circles is bounded from below by a term exponential in the degrees of the two vertices [16]. Since the sum of degrees in a planar graph is linear, the worst-case area ratio along any path in the graph is exponential in the graph size and the bound follows.

6 Conclusions

We have shown that, in confluent drawing, restricting attention to strict drawings allows us to characterize their combinatorial complexity. The recognition problem, whether a graph admits a strict confluent drawing has been shown to be NP-complete, but we have also shown that outerplanar strict confluent drawings with a fixed vertex ordering may be constructed in polynomial time.

The most pressing problem left open by this research is to recognize the graphs that have outerplanar strict confluent drawings, without imposing a fixed vertex order. Can we recognize these graphs in polynomial time? Or which other restrictions of strict confluent drawings admit a polynomial time recognition? Moreover, it is interesting to study the complexity of the recognition problem for strict confluent drawings of restricted graph classes. For outerplanar strict confluent drawings we gave a construction that uses at most two circular arcs per confluent arc. An interesting question is whether sublinear or constant bounds on the number of circular arcs per edge (as a sequence of confluent arcs) can be given. Last but not least, the complexity of the general question whether a given graph has a (not necessarily strict) confluent drawing is also still open.

Acknowledgements. This work originated at Dagstuhl seminar 13151, *Drawing Graphs and Maps with Curves*. D. Eppstein was supported in part by the National Science Foundation under grants 0830403 and 1217322, and by the Office of Naval Research under MURI grant N00014-08-1-1015. M. Löffler was supported by the Netherlands Organisation for Scientific Research (NWO) under grant 639.021.123. M. Nöllenburg received financial support by the 'Concept for the Future' of KIT under grant YIG 10-209. B. Speckmann is supported by NWO under project no. 639.023.208. K. Verbeek is supported by NWO under project no. 639.021.541.

References

[1] Michael A. Bekos, Michael Kaufmann, Stephen G. Kobourov, and Antonios Symvonis. Smooth orthogonal layouts. In *Graph Drawing 2012*, volume 7704 of *LNCS*, pages 150–161. Springer, 2013.

[2] Charles R. Collins and Kenneth Stephenson. A circle packing algorithm. *Computational Geometry Theory and Applications*, 25(3):233–256, 2003.

[3] Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li. Geometry-based edge clustering for graph visualization. *IEEE Trans. Visualization Computer Graphics*, 14(6):1277–84, 2008.

[4] Matthew Dickerson, David Eppstein, Michael T. Goodrich, and Jeremy Y. Meng. Confluent drawings: Visualizing non-planar diagrams in a planar way. *J. Graph Algorithms and Applications*, 9(1):31–52, 2005.

[5] Tim Dwyer, Kim Marriott, and Michael Wybrow. Integrating edge routing into force-directed layout. In *Graph Drawing 2006*, volume 4372 of *LNCS*, pages 8–19. Springer, 2007.

[6] David Eppstein, Michael T. Goodrich, and Jeremy Y. Meng. Confluent layered drawings. *Algorithmica*, 47(4):439–452, 2007.

[7] David Eppstein, Michael T. Goodrich, and Jeremy Yu Meng. Delta-confluent drawings. In *Graph Drawing 2005*, volume 3843 of *LNCS*, pages 165–176. Springer, 2006.

[8] David Eppstein, Danny Holten, Maarten Löffler, Martin Nöllenburg, Bettina Speckmann, and Kevin Verbeek. Strict confluent drawing. In *Graph Drawing 2013*, volume 8242 of *LNCS*, pages 352–363. Springer, 2013.

[9] David Eppstein and Joseph A. Simons. Confluent Hasse diagrams. In *Graph Drawing 2011*, volume 7034 of *LNCS*, pages 2–13. Springer, 2012.

[10] Michael Hirsch, Henk Meijer, and David Rappaport. Biclique edge cover graphs and confluent drawings. In *Graph Drawing 2006*, volume 4372 of *LNCS*, pages 405–416. Springer, 2007.

[11] Danny Holten. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Trans. Visualization Computer Graphics*, 12(5):741–8, 2006.

[12] Danny Holten and Jarke J van Wijk. Force-Directed Edge Bundling for Graph Visualization. *Computer Graphics Forum*, 28(3):983–990, 2009.

[13] Peter Hui, Michael J. Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Train tracks and confluent drawings. *Algorithmica*, 47(4):465–479, 2007.

[14] C. Hurter, O. Ersoy, and Alexandru Telea. Graph Bundling by Kernel Density Estimation. *Computer Graphics Forum*, 31(3pt1):865–874, 2012.

[15] David Lichtenstein. Planar formulae and their uses. *SIAM J. Computing*, 11(2):329–343, 1982.

[16] Seth Malitz and Achilleas Papakostas. On the angular resolution of planar graphs. *SIAM J. Discrete Mathematics*, 7(2):172–183, 1994.

[17] Kurt Mehlhorn and Stefan Näher. From algorithms to working programs: On the use of program checking in LEDA. In *Mathematical Foundations of Computer Science*, LNCS 1450, pages 84–93, 1998.

[18] Bojan Mohar. A polynomial time circle packing algorithm. *Discrete Mathematics*, 117(1–3):257–263, 1993.

[19] Helen Purchase. Which aesthetic has the greatest effect on human understanding? In *Graph Drawing (GD'97)*, volume 1353 of *LNCS*, pages 248–261. Springer, 1997.

[20] Gianluca Quercini and Massimo Ancona. Confluent drawing algorithms using rectangular dualization. In *Graph Drawing 2010*, volume 6502 of *LNCS*, pages 341–352. Springer, 2011.

[21] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Systems, Man, and Cybernetics*, 11(2):109–125, 1981.