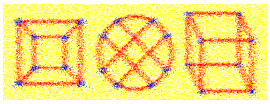

Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 28, no. 1, pp. 275–299 (2024)
DOI: 10.7155/jgaa.v28i1.2950

On Upward-Planar L-Drawings of Graphs

Patrizio Angelini[1] *Steven Chaplick*[2] *Sabine Cornelsen*[3] *Giordano Da Lozzo*[4]

[1]John Cabot University, Rome, Italy
[2]Maastricht University, The Netherlands
[3]University of Konstanz, Germany
[4]Roma Tre University, Rome, Italy

Submitted: April 2023 Accepted: April 2024 Published: July 2024

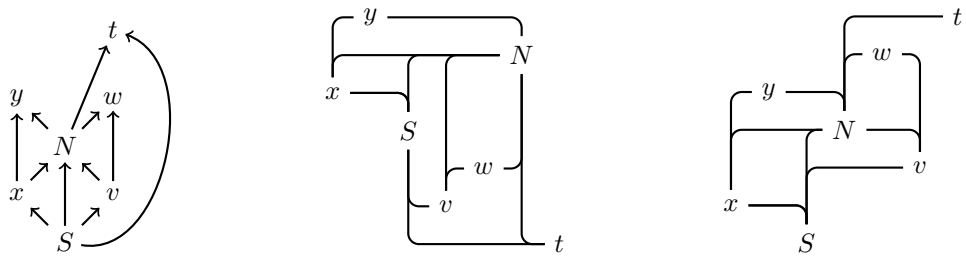
Article type: Regular paper Communicated by: I. Rutter

Abstract. In an *upward-planar L-drawing* of a directed acyclic graph (DAG) each edge $e = (v, w)$ is represented as a polyline composed of a vertical segment with its lowest endpoint at the *tail* v of e and of a horizontal segment ending at the *head* w of e. Distinct edges may overlap, but must not cross. Recently, upward-planar L-drawings have been studied for *st*-graphs, i.e., planar DAGs with a single source s and a single sink t containing an edge directed from s to t. It is known that a *plane st-graph*, i.e., an embedded *st*-graph in which the edge (s, t) is incident to the outer face, admits an upward-planar L-drawing if and only if it admits a bitonic *st*-ordering, which can be tested in linear time.

We study upward-planar L-drawings of DAGs that are not necessarily *st*-graphs. As a combinatorial result, we show that a plane DAG admits an upward-planar L-drawing if and only if it is a subgraph of a plane *st*-graph admitting a bitonic *st*-ordering. This allows us to show that not every tree with a fixed bimodal embedding admits an upward-planar L-drawing. Moreover, we prove that any directed acyclic cactus with a single source (or a single sink) admits an upward-planar L-drawing, which respects a given outerplanar embedding if there are no transitive edges. On the algorithmic side, we consider DAGs with a single source (or a single sink). We give linear-time testing algorithms for these DAGs in two cases: (a) when the drawing must respect a prescribed embedding and (b) when no restriction is given on the embedding, but the underlying undirected graph is series-parallel. For the single-sink case of (b) it even suffices that each biconnected component is series-parallel.

Da Lozzo was supported, in part, by MUR of Italy (PRIN Project no. 2022ME9Z78 – NextGRAAL and PRIN Project no. 2022TS4Y3N – EXPAND). A preliminary version of this paper appeared in [2].

E-mail addresses: pangelini@johncabot.edu (Patrizio Angelini) s.chaplick@maastrichtuniversity.nl (Steven Chaplick) sabine.cornelsen@uni-konstanz.de (Sabine Cornelsen) giordano.dalozzo@uniroma3.it (Giordano Da Lozzo)



(a) DAG (b) Planar L-drawing (c) Upward-planar L-drawing

Figure 1: (a) A single-source series-parallel DAG G with poles S and N. (b) A planar L-drawing of G. (c) An upward-planar L-drawing of G without the edge $\{S, t\}$.

1 Introduction

In order to visualize hierarchies, directed acyclic graphs (DAGs) are often drawn in such a way that the geometric representation of the edges reflects their direction. To this aim, *upward drawings* have been introduced, i.e., drawings in which edges are monotonically increasing curves in the y-direction if traversed from tail to head. Sugiyama et al. [24] provided a general framework for drawing DAGs in an upward way. To support readability, it is desirable to avoid edge crossings [23, 26]. However, not every planar DAG admits an *upward-planar drawing*, i.e., an upward drawing in which no two edges intersect except in common endpoints. A necessary condition is that the corresponding embedding is *bimodal*, i.e., all incoming edges are consecutive in the cyclic sequence of edges around any vertex. Di Battista and Tamassia [15] showed that a DAG is upward-planar if and only if it is a subgraph of a *planar st-graph*, i.e., a planar DAG with a single source and a single sink that are connected by an edge. Based on this characterization, it can be decided in near-linear time whether a DAG admits an upward-planar drawing respecting a given planar embedding [6, 9]. However, it is $\mathcal{NP}$-complete to decide whether a DAG admits an upward-planar drawing when no fixed embedding is given [19]. For special cases, upward-planarity testing in the variable embedding setting can be performed in polynomial time: e.g. if the DAG has only one source [7, 10, 22], or if the underlying undirected graph is a partial 2-tree [13, 17], i.e. the biconnected components are series-parallel. See Fig. 1a for an example of a DAG that is both single-source and series-parallel. Furthermore, parameterized algorithms for upward-planarity testing exist with respect to the number of sources or the treewidth of the input DAG [12].

Every upward-planar DAG admits a straight-line upward-planar drawing [15], however such a drawing may require exponential area [16]. The class of plane st-graphs admitting an upward-planar drawing in quadratic area include those that also admit a *bitonic st-ordering* [20], i.e., an enumeration of the vertices such that edges point from lower numbers to greater numbers and such that the left-to-right successor list of each vertex is enumerated first ascending and then descending. It can be tested in linear time whether a plane st-graph admits a bitonic st-ordering [20], and whether a planar st-graph admits a planar embedding that allows for a bitonic st-ordering [1, 11]. By subdividing some transitive edges once, every plane st-graph can be extended such that it admits a bitonic st-ordering. Moreover, the minimum number of edges that have to be subdivided can be

determined in linear time, both in the variable [1] and in the fixed embedding setting [20]. This implies in particular that each upward-planar graph has an upward-planar drawing in quadratic area with at most one bend per edge.

In an *L-drawing* of a directed graph [5] each edge e is represented as a polyline composed of a vertical segment incident to the tail of e and a horizontal segment incident to the head of e. In a *planar L-drawing*, distinct edges may overlap, but not cross. See Fig. 1b for an example. The problem of testing for the existence of a planar L-drawing of a directed graph is $\mathcal{NP}$-complete [11]. On the other hand, every upward-planar DAG admits a planar L-drawing [4]. A planar L-drawing is *upward* if the lowest end vertex of the vertical segment of every edge e is the tail of e. See Fig. 1c for an example. A planar st-graph admits an upward-planar L-drawing if and only it admits a bitonic st-ordering [11]. *Planar confluent orthogonal drawings*, which are L-drawings of graph subdivisions, are considered in [14].

Our Contribution. In Section 3, we characterize the plane DAGs admitting an upward-planar L-drawing as the subgraphs of plane st-graphs admitting a bitonic st-ordering (Section 3). Together with [20], this implies that plane DAGs admitting an upward-planar L-drawing also admit an upward-planar straight-line drawing in quadratic area. We first apply this characterization to prove that there are trees with a fixed bimodal embedding that do not admit an upward-planar L-drawing (Section 3). Moreover, the characterization allows us to test in linear time whether any DAG with a single source or a single sink admits an upward-planar L-drawing preserving a given embedding (Section 4.2) in Section 4.

In Section 4, we further show that every single-source acyclic cactus admits an upward-planar L-drawing by directly computing the x- and y-coordinates as post- and pre-order numbers, respectively, in a DFS-traversal (Section 4.1). The respective result holds for single-sink acyclic cacti. Finally, in Section 5, we give a dynamic-programming approach to decide in linear time whether (i) a (biconnected) series-parallel DAG with a single source or (ii) a DAG with a single sink where each biconnected component is series-parallel has an embedding admitting an upward-planar L-drawing (Section 5). To this end, we characterize the L-drawings of the different components by a constant set of regular expressions of constant length, i.e., independent of the size of the DAG. Observe that a plane st-graph does not necessarily admit an upward-planar L-drawing if the st-graph obtained by reversing the orientation of its edges does. This justifies studying single-source and -sink graphs independently.

2 Preliminaries

For standard graph theoretic notations and definitions we refer the reader to [25].

Digraphs. A *directed graph (digraph)* $G = (V, E)$ is a pair consisting of a finite set V of *vertices* and a set E of edges containing ordered pairs of distinct vertices. A vertex of a digraph is a *source* if it is only incident to outgoing edges and a *sink* if it is only incident to incoming edges. A *walk* is a sequence of vertices such that any two consecutive vertices in the sequence are adjacent. A *path* is a walk with distinct vertices. In this work we assume that all graphs are *connected*, i.e., that there is always a path between any two vertices. A *cycle* is a walk with distinct vertices except for the first and the last vertex which must be equal. A *directed path* (*directed cycle*) is a path (cycle) where for any vertex v and its successor u in the path (cycle) there is an edge directed from u to v. In the following, we only consider *acyclic digraphs (DAGs)*, i.e., digraphs that do not contain directed cycles. A DAG is a *tree* if it is connected and contains no cycles. It is a *cactus* if it is connected and each edge is contained in at most one cycle.

Drawings. In a *drawing (node-link diagram)* of a digraph vertices are drawn as points in the plane and edges are drawn as simple curves between their end vertices. A drawing of a DAG is *planar* if no two edges intersect except in common endpoints. A planar drawing splits the plane into connected regions – called *faces*. A *planar embedding* of a DAG is the counter-clockwise cyclic order of the edges around each vertex according to a planar drawing. A *plane* DAG is a DAG with a fixed planar embedding and a fixed unbounded face. A planar embedding of a DAG is *bimodal* if all incoming edges are consecutive in the cyclic sequence of edges around any vertex, i.e. if in the cyclic sequence of edges around any vertex v there are at most two pairs of consecutive edges that are neither both incoming nor both outgoing.

In general, the *rotation* of a polygonal chain is the sum over the angular deviations from a straight line at each bend, where counterclockwise counts positive and clockwise counts negative. More specifically, the *rotation* of an orthogonal polygonal chain, possibly with overlapping edges, is defined as follows: We start with rotation zero. If the curve bends to the left, i.e., if there is a convex angle to the left of the curve, then we add $\pi/2$ to the rotation. If the curve bends to the right, i.e., if there is a concave angle to the left of the curve, then we subtract $\pi/2$ from the rotation. Moreover, if the curve has a 2π angle to the left, we handle this as if bending twice to the right, and if there is a 0 angle to the left, we handle this as if bending twice to the left. The rotation of a simple polygon – with possible overlaps of consecutive edges – traversed in counterclockwise direction is 2π.

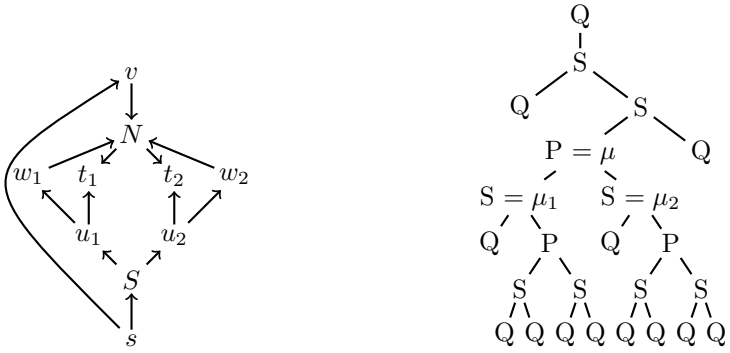
Planar *st*-graphs and bitonic *st*-ordering. A *planar st-graph* is a planar DAG with a single source s, a single sink t, and an edge (s,t). An *st-ordering* of a planar st-graph is an enumeration π of the vertices with distinct integers, such that $\pi(u) < \pi(v)$ for every edge (u,v). A *plane st-graph* is a planar st-graph with a planar embedding in which the edge (s,t) is incident to the outer face. Every plane st-graph admits an upward-planar drawing [15].

For each vertex v of a plane st-graph, we consider the ordered list $S(v) = \langle v_1, v_2, \ldots, v_k \rangle$ of the successors of v as they appear from left to right in an upward-planar drawing. An st-ordering of a plane st-graph is *bitonic*, if there is a vertex v_h in $S(v) = \langle v_1, v_2, \ldots, v_k \rangle$ such that $\pi(v_i) < \pi(v_{i+1})$, $i = 1, \ldots, h-1$, and $\pi(v_i) > \pi(v_{i+1})$, $i = h, \ldots, k-1$. We say that the successor list $S(v) = \langle v_1, v_2, \ldots, v_k \rangle$ of a vertex v contains a *valley* if there are $1 < i \leq j < k$ such that there are both, a directed v_i-v_{i-1}-path and a directed v_j-v_{j+1}-path in G. See Fig. 5. Gronemann [20] characterized the plane st-graphs that admit a bitonic st-ordering as follows.

Theorem 1 ([20]) *A plane st-graph admits a bitonic st-ordering if and only if the successor list of no vertex contains a valley.*

Series-parallel DAGs. A *series-parallel digraph* is a digraph whose underlying undirected graph is *series-parallel*, i.e., a graph with two distinguished vertices, called *poles*, which can be defined recursively as follows: A single edge is a series-parallel graph. Given two series-parallel graphs G_1 and G_2 (*components*), with poles v_i, u_i, $i = 1, 2$, a series-parallel graph G with poles v and u can be obtained in two ways: by merging v_1, v_2 and u_1, u_2, respectively, into the new poles v and u (*parallel composition*), or by merging the vertices u_2 and v_1 and setting $u = u_1$ and $v = v_2$ (*series composition*). We remark that with this definition series-parallel graphs are always biconnected after the addition of an edge between the poles.

On the other hand, given a biconnected series-parallel DAG G with an edge e between the poles, we represent the recursive construction of the graph $G - e$, which is obtained by removing e from G, in a *binary decomposition tree* T. We refer to the vertices of T as *nodes*. The leaves (nodes of degree one) of T are labeled Q and represent the edges. The other nodes (*inner nodes*) are



(a) DAG not upward-planar (b) Binary decomposition tree

Figure 2: a) A bimodal single-source series-parallel DAG that is not upward-planar b) with its decomposition tree.

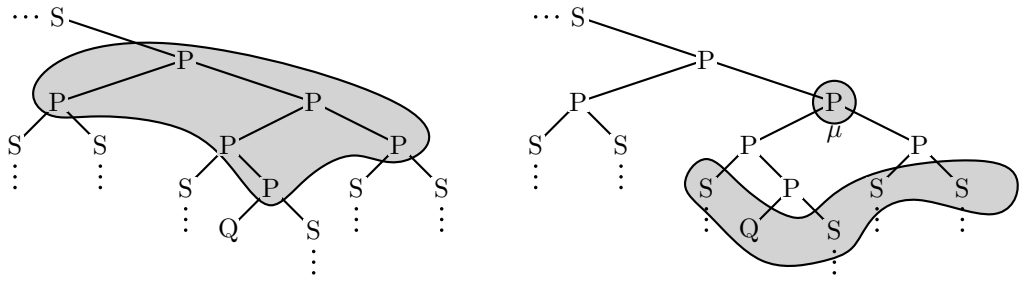
(a) Embedding not upward (b) Embedding upward

Figure 3: A single-source series-parallel DAG with (a) an embedding that is not upward-plane (b) a different embedding that is upward-planar.

labeled P for parallel composition or S for series composition. Finally, we consider T rooted at an additional Q-node representing the edge e, which connects the poles of its unique child component. Fig. 2b shows a binary decomposition tree of the graph in Fig. 2a.

The *canonical decomposition tree* of G is obtained from the binary decomposition tree by merging maximal connected components of P- or S-nodes, respectively, into a single P- or S-node. See Fig. 4a. Observe that in the canonical decomposition tree T no two adjacent nodes of T have the same label. While different orderings of the children of an S-node represent different graphs, the different permutations of the children of a P-node in the canonical decomposition tree represent different planar embeddings of the same series-parallel graph. We say that a P-node or S-node, respectively, of the canonical decomposition tree represents a parallel or series composition of ℓ components if it has ℓ children. Moreover, the *canonical descendants* of a node μ of the binary decomposition tree are the descendants μ' of μ such that μ' has a different type than μ and the nodes in the unique path between μ and μ' have the same type as μ. See Fig. 4b.

Let μ be a node of a decomposition tree T and let ν be a neighbor of μ. We denote by $T_\nu(\mu)$ the connected component of T without the edge $\{\mu, \nu\}$ that contains μ. Analogously, we denote by $G_\nu(\mu)$ the subgraph of G corresponding to $T_\nu(\mu)$, i.e., the subgraph of G formed by the edges corresponding to the leaves of $T_\nu(\mu)$. The poles of $G_\nu(\mu)$ are the two vertices shared by $G_\nu(\mu)$ and $G_\mu(\nu)$. The vertices of $G_\nu(\mu)$ that are different from its poles are called *internal*. We omit



(a) Connected Component of P-nodes (b) Canonical Descendants of μ

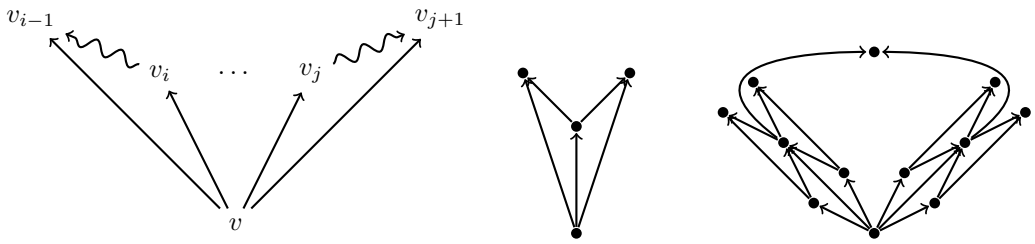
Figure 4: The canonical decomposition tree is obtained from a binary decomposition tree by merging maximal connected components of nodes of the same type.

the subscript ν from the notation if it is clear from the context. Given an arbitrary biconnected digraph G, it can be determined in linear time whether it is series-parallel, and a decomposition tree of G can be computed also in linear time [21]. Moreover, rooting a decomposition tree of a biconnected series-parallel digraph G at an arbitrary Q-node yields again a decomposition tree of G, but with different poles. Observe that the choice of the root represents a planar embedding with the respective edge incident to the outer face.

In the following, we assume that G is a biconnected series-parallel DAG with a single source s or a single sink t. We root the decomposition tree T at a Q-node corresponding to an edge incident to s or t, respectively. Observe that $G(\mu) = G_\nu(\mu)$ where ν is the neighbor of μ on the path to the root. This implies that for any node μ of T no internal vertex of $G(\mu)$ can be a source (sink) of $G(\mu)$ and at least one of the poles of $G(\mu)$ is a source (sink) of $G(\mu)$.

It follows from [7] that every single-source series-parallel DAG is upward-planar if each vertex is incident to at most one incoming or at most one outgoing edge. However, even in that case, not every bimodal embedding is already upward-planar, see Fig. 3a. Moreover, not every single-source series-parallel DAG is upward-planar, even if it admits a bimodal embedding, see Fig. 2a. The reason for that is a P-node μ with two children μ_1 and μ_2 such that a pole N of $G(\mu)$ is incident to an incoming edge in $G - G(\mu)$, and to both incoming and outgoing edges in both, $G(\mu_1)$ and $G(\mu_2)$. (No matter how we would permute the edges of $G(\mu_1)$ and $G(\mu_2)$ around N, there would always be an outgoing edge of N that is trapped between two incoming edges of N.) Bimodal single-source series-parallel DAGs without this property are always upward-planar [3].

Given an upward-planar drawing of G with distinct y-coordinates for the vertices, we call the pole of $G(\mu)$ with lower y-coordinate the *South pole* of $G(\mu)$ and the other pole the *North pole* of $G(\mu)$. Observe that in the single-source case the South pole of G is the unique source s and the North pole does not have to be a sink; see Fig. 1a. In the single-sink case, the North-pole is the unique sink. If μ is a P-node with children $\mu_1, \ldots, \mu_\ell$, then the South pole of $G(\mu_i)$, $i = 1, \ldots, \ell$ is the South pole of $G(\mu)$. Finally, if μ is an S-node with children $\mu_1, \ldots, \mu_\ell$, then observe that at most one among the components $G(\mu_i)$, $i = 1, \ldots, \ell$ can have more than one source (sink) – otherwise G would have more than one source (sink). The South (North) pole of all other components is their unique source (sink).



(a) Valley (b) Fixed (c) Variable

Figure 5: (a) Forbidden configuration for bitonic st-orderings. (b+c) Single-source series-parallel plane DAG that does not admit an upward-planar L-drawing since it contains a valley, in case (c) in any upward-planar embedding.

3 Upward-Planar L-Drawings – A Characterization

A plane st-graph admits an upward-planar L-drawing if and only it admits a bitonic st-ordering [11]. We extend this result to general plane DAGs and discuss some consequences.

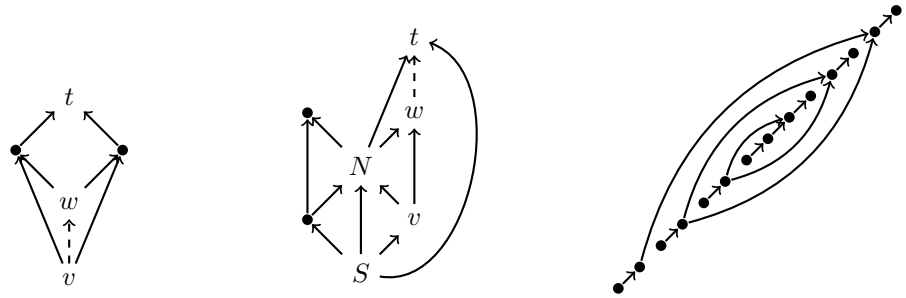
Theorem 2 (Relationship to bitonic st-orderings) *A plane DAG admits an upward-planar L-drawing if and only if it can be augmented to a plane st-graph that admits an upward-planar L-drawing, i.e., a plane st-graph that admits a bitonic st-ordering.*

Proof: Let G be a plane DAG. Clearly, if an augmentation of G admits an upward-planar L-drawing, then so does G. Let now an upward-planar L-drawing of G be given. Add a directed triangle with a new source s, a new sink t, and a new vertex x enclosing the drawing of G. As long as there is a vertex v of G that is not incident to an incoming or outgoing edge, shoot a ray from v to the right or the top, respectively, until it hits another edge and follow the segment to the incident vertex – recall that one end of any segment is a vertex and one end is a bend. The orientation of the added edge is implied by the L-drawing. The result is an upward-planar L-drawing of a digraph with the single source s and the single sink t. □

Observe that every series-parallel st-graph admits a bitonic st-ordering [1, 11] and, thus, an upward-planar L-drawing. This is no longer true for upward-planar series-parallel DAGs with several sources or several sinks. Figs. 5b and 5c show examples of two single-source upward-planar series-parallel DAGs that contain a valley. There are even upward-planar series-parallel DAGs with a single source or a single sink that do not admit an upward-planar L-drawing, even though the successor list of no vertex contains a valley.

Consider the DAG G in Fig. 6a (without the dashed edge). G has a unique upward-planar embedding up to symmetry. Since no vertex has more than two successors there cannot be a valley. Assume G admits a planar L-drawing. By Section 3 there should be an extension of G to a plane st-graph G' that admits a bitonic st-ordering. But the internal source w can only be eliminated by adding the edge (v, w). Thus w is a successor of v in G'. Hence, the successor list of v in G' contains a valley. By Section 2, G' is not bitonic, a contradiction.

Now consider the DAG G in Fig. 6b (without the dashed edge). G has two symmetric upward-planar embeddings: with the curved edge to the right or the left of the remainder of the DAG. Suppose G admits a planar L-drawing. We may assume that the curved edge is to the right. But then an augmentation to a plane st-graph G' must contain the dashed edge or its reversal, which



(a) Single sink (b) Single source (c) Family of trees

Figure 6: DAGs that do not admit an upward-planar L-drawing even though they do not contain a valley. Dashed edges indicate augmentations and are not part of the DAG.

completes a valley at the single source and its three rightmost outgoing edges. By Section 2, G' is not bitonic, a contradiction.

A planar L-drawing is *upward-leftward* [11] if all edges are upward and point to the left.

Theorem 3 (Trees) *Every directed tree admits an upward-leftward planar L-drawing, but not every tree with a fixed bimodal embedding admits an upward-planar L-drawing.*

Proof: If the embedding is not fixed, we can construct an upward-planar L-drawing of the input tree by removing one leaf v and its incident edge e, drawing the smaller directed tree inductively, and inserting the removed leaf into this upward-leftward planar L-drawing. To this end let u be the unique neighbor of v. We embed e as the first incoming or outgoing edge of u, respectively, in counterclockwise direction, and draw v slightly to the right and below u, if e is an incoming edge of u, or slightly to the left and above u, if e is an outgoing edge of v. This guarantees that the resulting L-drawing is upward-leftward and planar.

When the embedding is fixed, we consider a family of plane trees T_k, $k \geq 1$, proposed by Frati [18, Fig. 4a], that have $2k$ vertices and require an exponential area $\Omega(2^{k/2})$ in any embedding-preserving straight-line upward-planar drawing; see Fig. 6c. We claim that, for sufficiently large k, the tree T_k does not admit an upward-planar L-drawing. Suppose, for a contradiction, that it admits one. By Section 3, we can augment this drawing to an upward-planar L-drawing of a plane st-graph G with $n = 2k+3$ vertices. This implies that G admits a bitonic st-ordering [11]. Hence, G (and thus T_k) admits a straight-line upward-planar drawing in quadratic area $(2n-2) \times (n-1)$ [20], a contradiction. □

4 Single-Source or -Sink DAGs with Fixed Embedding

In the fixed embedding scenario, we first prove that every single-source or -sink acyclic cactus with no transitive edge admits an upward-planar L-drawing and then give a linear-time algorithm to test whether a single-source or -sink DAG admits an upward-planar L-drawing.

4.1 Cacti

Theorem 4 (Plane Single-Source or Single-Sink Cacti) *Every acyclic cactus G with a single source or single sink admits an upward-leftward outerplanar L-drawing. Moreover, if there are*

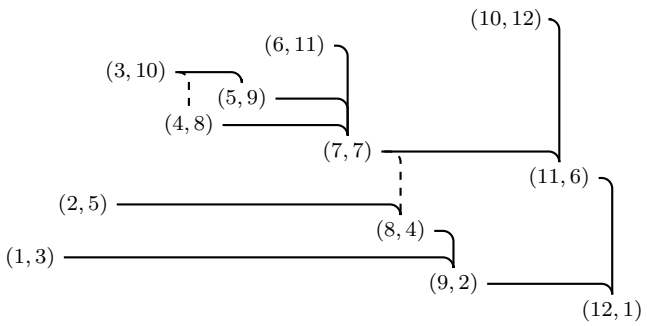


Figure 7: Single-source acyclic cactus. The dashed edges are the last edges on a left path of some cycles.

no transitive edges (e.g., if G is a tree) then such a drawing can be constructed so to maintain a given outerplanar embedding.

Proof: Let G be an acyclic cactus. We first consider the case that G has a single source s. Observe that then each biconnected component C of G (which is either an edge or a cycle) has a single source, namely the cut-vertex of G that separates it from the part of the DAG containing s. This implies that C also has a single sink (although G may have multiple sinks, belonging to different biconnected components). In particular, if C is a cycle, it consists of a *left path* P_ℓ and a *right path* P_r between its single source and single sink. By flipping the cycle C – maintaining outerplanarity – we can ensure that P_ℓ contains more than one edge. Note that this flipping is only performed if there are transitive edges. Consider the tree T that results from G by removing the last edge of every left path.

We perform a depth-first search on T starting from s where the edges around a vertex are traversed in clockwise order. We enumerate each vertex twice, once when we first meet it (DFS-number or *preorder* number) and once when we finally backtrack from it (*postorder* number). To also obtain that each edge points to the left, backtracking has to be altered from the usual DFS: Before backtracking on a left path P_ℓ of a cycle C, we directly jump to the single source s_C of C and continue the DFS from there, following the right path P_r of C. Only once we have backtracked from the single sink t_C of C, we give each vertex on P_ℓ, excluding s_C, a postorder number and then we continue backtracking on P_r. See Fig. 7.

Let the y-coordinate of a vertex be its preorder number and let the x-coordinate be its thus constructed postorder number. Since each vertex has a larger preorder- and a lower postorder-number than its parent, we draw every edge (u, v) with a vertical segment exiting u from above, a horizontal segment entering v from the right, thus, the drawing is upward-leftward. We now prove in three steps that the thus constructed drawing is planar and preserves the embedding. Observe that the embedding was updated only in the presence of transitive edges.

1. The drawing of T preserves the given embedding: Each vertex has at most one incoming edge since we have removed the last edge of every left path. Let (v, w_1) and (v, w_2) be two outgoing edges of a vertex v such that (v, w_1) is to the left of (v, w_2) in the order of the outgoing edges of v. By outerplanarity, the children of v are traversed from left to right. It follows that w_1 has a lower preorder-number (y-coordinate) than w_2.

2. The drawing of T is planar: Let v be a vertex and let w_1 and w_2 be two children of a vertex v

such that w_1 is to the left of w_2. Since the children of v are traversed from left to right it follows that the vertices in subtree T_1 rooted at w_1 obtain a lower preorder-number than the vertices in the subtree T_2 rooted at w_2. Thus, the bounding box of T_1 is below the bounding box of T_2. Hence, no edge of T_1 can intersect an edge of T_2. Further, w_1 and w_2 are the rightmost points in T_1 and T_2, respectively, and v is to the right of both T_1 and T_2. Thus, edges incident to v cannot cross edges within T_1 or T_2. Recursively, we obtain that there are no crossings.

3. The drawing of G is planar and preserves its embedding. Let now (u, t_C) be the last edge on the left path of a cycle C. Recall that (u, t_C) is not a transitive edge. Thus, by the special care we took for the backtracking, we know that the x-coordinates of u and t_C differ only by one. This implies that (u, t_C) is the leftmost incoming edge of t_C. Moreover, (u, t_C) could at most be crossed by a horizontal segment of an edge in T. However, the vertices with y-coordinate between the y-coordinate of u and the y-coordinate of t_C are either in the subtree T_u rooted at u, and thus to the left of u, or they are on the right path P_r of C and, thus to the right of u. Since there are no edges between vertices in T_u and P_r, it follows that (u, t_C) is not crossed. Moreover, (u, t_C) is the rightmost outgoing edge of u.

Now consider the case that G has a single sink. Flip the embedding, i.e., reverse the linear order of the incoming (outgoing) edges around each vertex. Reverse the orientation of the edges, construct the drawing of the resulting single-source DAG, rotate it by 90 degrees in counter-clockwise direction, and mirror it horizontally. This yields the desired drawing. □

4.2 General Single-Source or Single-Sink DAGs

Next we consider general DAGs with a single source or a single sink and a fixed embedding. We show how to test in linear time whether such graphs admit an upward-planar L-drawing. We first introduce some notation and tools and use these to prove Section 4.2.

Two consecutive incident edges of a vertex form an *angle*. A *large angle* in an upward-planar straight-line drawing is an angle greater than π between two consecutive edges incident to a source or a sink, respectively. An *upward-planar embedding* of an upward-planar DAG is a planar embedding with the assignment of large angles according to a straight-line upward-planar drawing. For single-source or single-sink DAGs, respectively, a planar embedding and a fixed outer face already determine an upward-planar embedding [7].

An angle is a *source-switch* or a *sink-switch*, respectively, if the two edges are both outgoing or both incoming edges of the common end vertex. Observe that the number $A(f)$ of source-switches in a face f equals the number of sink-switches in f. Bertolazzi et al. [6] proved that in biconnected upward-planar DAGs, the number $L(f)$ of large angles in a face f is $A(f) - 1$, if f is an inner face, and $A(f) + 1$, otherwise, and mentioned in the conclusion that this result could be extended to simply connected graphs. For completeness, we now provide an explicit proof for single-source or -sink DAGs.

Lemma 1 *In a plane single-source or -sink DAG, the number $L(f)$ of large angles in a face f is $A(f) - 1$, if f is an inner face, and $A(f) + 1$, otherwise.*

Proof: By symmetry, it suffices to consider an upward-planar DAG G with a single source s and with a fixed upward-planar drawing. Observe that one angle at s in the outer face is the only large angle of G at a source-switch. We do induction on the number of biconnected components.

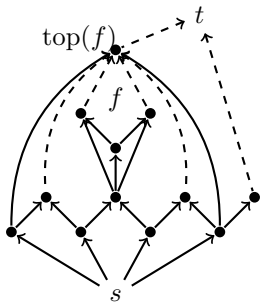


Figure 8: A new sink t and the dashed edges augment a plane single-source DAG to a plane st-graph.

If G is biconnected the statement follows from [6]. So let C be a biconnected component of G that contains exactly one cut vertex v and such that $C-v$ does not contain the single source s. Observe that then v is the single source of C: Indeed on one hand if C had another source then this would be also a source of G. On the other hand, C is a DAG and, thus, has a source. Additionally, we may assume that the interior of C does not contain any other vertex of G. Let G' be the DAG obtained from G by removing C but not v. Similarly as for v, the vertex s is the single source of G'. By the inductive hypothesis, G' and C have the required property. Now consider the face f' of G' containing C, let f_0 be the outer face of C, and let f be the face in G contained in both, f' and f_0.

Since v is a source in C, all sink-switches of C are still sink-switches of G. Assume first that the angle at v in f' was not a sink-switch of G'. Then the sink-switches of G' are still sink-switches of G. Hence, in this case, we have that $A(f) = A(f') + A(f_0)$. Moreover, the large angles at sink-switches of G in f are the large angles at sink-switches of G' in f' plus the large angles at sink-switches of C in f_0. The source-switch in f_0 at the source v of C is large, since f_0 is the outer face of C. However, if f is an inner face, then the source-switch at v in f cannot be large. Thus $L(f) = L(f') + L(f_0) - 1 = A(f') - 1 + A(f_0) + 1 - 1 = A(f) - 1$. If f is the outer face, then both, C and G' had a large angle at a source-switch in the outer face, but G has only one. We obtain $L(f) = L(f') + L(f_0) - 1 = A(f') + 1 + A(f_0) + 1 - 1 = A(f) + 1$. Finally, if C is contained in a sink-switch angle of G' at v then this must be a large angle and we get $A(f) = A(f') - 1 + A(f_0)$ and $L(f) = L(f') - 1 + L(f_0) - 1 = A(f') \pm 1 - 1 + A(f_0) + 1 - 1 = A(f) \pm 1$, where the ± 1 distinguishes between the outer and the inner face. □

Theorem 5 *Given an upward-plane single-source or single-sink DAG, it can be tested in linear time whether it admits an upward-planar L-drawing.*

Single Source. We first prove the theorem for a DAG G with a single source s. In an upward-planar straight-line drawing of G, the only large angle at a source-switch is the angle at s in the outer face. Thus, by Section 4.2, in the outer face all angles at sink-switches are large and in an inner face f all but one angle at sink-switches are large. For an inner face f, let top(f) be the sink-switch of f without large angle. See Fig. 8.

Lemma 2 *Let G be a single source upward-planar DAG with a fixed upward-planar embedding, let f be an inner face, and let v be a sink with a large angle in f. Every plane st-graph extending G contains a directed v-top(f)-path.*

Proof: Consider a planar st-graph extending G. In this graph there must be an outgoing edge e of v towards the interior of f. Let w be the head of e. Follow a path from w on the boundary of f upward until a sink-switch v' is met. If this sink-switch is $\text{top}(f)$, we are done. Otherwise continue recursively by considering an outgoing edge e' of v' toward the interior of f. Eventually this process terminates when $\text{top}(f)$ is reached. □

Proof: [Proof of Section 4.2, single-source case] Let G be an upward-planar single-source DAG with a fixed upward-planar embedding. Let G' be the DAG that results from G by adding in each inner face f edges from all sinks with a large angle in f to $\text{top}(f)$ and by adding a new sink t together with edges from all sink-switches on the outer face. We will show that G admits an upward-planar L-drawing if and only if G' does. This implies the statement, since testing whether G' admits a bitonic st-ordering can be performed in linear time [20].

Clearly, if $G' \supseteq G$ admits an upward-planar L-drawing, then so does G. In order to prove the other direction, suppose that G has an upward-planar L-drawing. In order to prove that G' admits an upward-planar L-drawing, we show that it is a planar st-graph that admits a bitonic st-ordering [11]. To show this, we argue that G' is acyclic, has a single source and a single sink, and the successor list of no vertex contains a valley by Section 2.

Namely, the edges to the new sink t cannot be contained in any directed cycle. Furthermore, by Section 3, there is an augmentation G'' of G such that (a) G'' is a planar st-graph and such that (b) G'' admits an upward-planar L-drawing. By Section 4.2, the edges added into inner faces of G either belong to G'' or are transitive edges in G''. Thus, G' is acyclic.

Since G' does not have more sources than G, there is only one source in G'. Each sink has a large angle in some face. Thus, in G' each vertex other than t has at least one outgoing edge. Therefore, G' is a planar st-graph.

Assume now that there is a face f with a sink w such that the edge $(w,\text{top}(f))$ would be part of a valley at a vertex v in G', i.e., assume there are successors $v_{i-1}, v_i, v_j, v_{j+1}$ of v from left to right (with possibly $v_i = v_j$) such that there is both, a directed v_i-v_{i-1}-path and a directed v_j-v_{j+1}-path. Since the out-degree of w in G' is one, it follows that $w \neq v$. Thus, $(w,\text{top}(f))$ could only be part of the v_i-v_{i-1}-path or the v_j-v_{j+1}-path. But then, by Section 4.2, there would be such a path in any augmentation of G to a planar st-graph, which, by Section 3, contradicts the fact the G admits an upward-planar L-drawing.

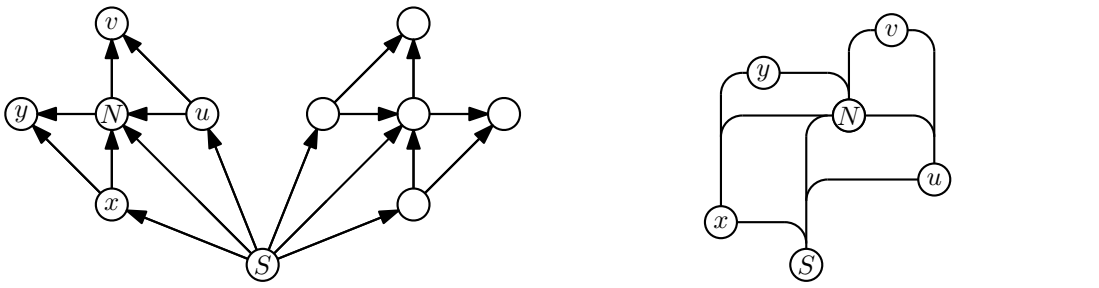
Finally, the edges incident to t cannot be involved in any valley, since all the tails have out-degree 1. Thus, G' contains no valleys. □

Single-Sink. The results for single-source DAGs translate to single-sink DAGs. Each inner face f has exactly one source-switch v that is not large which we call $\text{bottom}(f)$. With the analogous proof as for Section 4.2, we obtain.

Lemma 3 *Let G be a single sink upward-planar DAG with a fixed upward-planar embedding, let f be an inner face, and let v be a source with a large angle in f. Every plane st-graph extending G contains a directed bottom(f)-v-path.*

Proof: [Proof of Section 4.2, single-sink case] Let G be an upward-planar single-sink DAG. Augment each inner face f by edges from $\text{bottom}(f)$ to the incident source-switches with large angle. Add a new source s together with edges from all source-switches on the outer face.

Analogously as in the previous section, the thus constructed DAG has a single sink, a single source, and is acyclic.



(a) single-source DAG (b) Upward-Planar L-Drawing of One Component

Figure 9: A single-source DAG that has no upward-planar L-drawing even though each biconnected component does.

The edges incident to s cannot be contained in a valley since their heads have all in-degree one. By upward-planarity, the new edges incident to bottom(f) for some face f cannot be the transitive edges of a valley. Finally, if an edge incident to bottom(f) would be contained in a path then this path is unavoidable due to Section 4.2.

Thus, we have a planar st-graph G' that admits an upward-planar L-drawing if and only if the original DAG G did. It can be tested in linear time whether G' admits a bitonic st-ordering and thus an upward-planar L-drawing [20]. □

5 Single-Source or -Sink Series-Parallel DAGs with Variable Embedding

The goal of this section is to prove the following theorem.

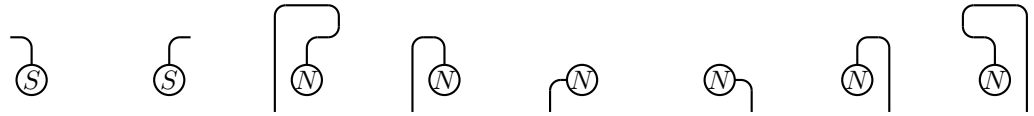
Theorem 6 *Let G be a DAG where each biconnected component is series-parallel. We can test in linear time whether G admits an upward-planar L-drawing,*

1. *if G has a single sink, or*
2. *if G is biconnected and has a single source.*

Lubiw and Hutton [22] showed that a single-source or single-sink DAG, respectively, is upward-planar if and only if all its biconnected components are. We show that for upward-planar L-drawings this is still true for single-sink DAGs, but in general not for single-source DAGs. See the DAG G in Fig. 9a which consists out of two identical biconnected components. An upward-planar L-drawing of this biconnected component is indicated in Fig. 9b. However, no matter how the embeddings of the two components are combined at S, there is always a valley.

Lemma 4 *A DAG with a single sink, whose embedding is not fixed, admits an upward-planar L-drawing if and only if all its biconnected components do.*

Proof: Let G be a DAG with a single sink t. Clearly, if G admits an upward-planar L-drawing, then so does every of its biconnected components. For the other direction, assume that all biconnected components of G admit an upward-planar L-drawing. We prove by induction on the number of



(a) L (b) R (c) R^{cc} (d) L^{cc} (e) W (f) E (g) R^{c} (h) L^{c}

Figure 10: The different types of a path between the poles. (a,b) South types; (c-h) North types.

biconnected components that G admits an upward-planar L-drawing. If G is biconnected there is nothing to show. Otherwise, let H be a biconnected component of G containing exactly one cut vertex v and such that $H - v$ does not contain t; this component exists since there exist at least two components with exactly one cut vertex. Note that v is the single sink of H. Let G' be the graph obtained from G by removing H, but not v. If $v \neq t$ then v cannot be a sink of G', otherwise v would be a sink of G. Thus, by the inductive hypothesis, G' and H have upward-planar L-drawings. We insert the edges incident to v in H between the bottommost incoming edge of G' entering v from the left and the bottommost incoming edge of G' entering v from the right (if any). This results in an upward-planar L-drawing of G. □

In the following we assume that G is a biconnected series-parallel DAG.

Single Source. We follow a dynamic-programming approach inspired by Binucci et al. [8] and Chaplick et al. [12]. We define feasible types that combinatorially describe the "shapes" attainable in an upward-planar L-drawing of each component. We show that these types are sufficient to determine the possible types of a graph obtained with parallel or series compositions, and show how to compute them efficiently. The types depend on the choice of the South pole as the bottommost pole (if it is not uniquely determined by the structure of the graph, e.g., if one of them is the unique source), and on the type of the leftmost S-N-path P_L and the rightmost S-N-path P_R between the South-pole S and the North-pole N. Observe that P_L and P_R do not have to be directed paths.

More precisely, the type of an S-N-path P is defined as follows: There are two *South-types* depending on the edge incident to S: L (outgoing edge bending to the *left*; Fig. 10a) and R (outgoing edge bending to the *right*; Fig. 10b). For the last edge incident to the North pole N we have in addition the types for the incoming edges: W (incoming edge entering from the left – *West*; Fig. 10e) and E (incoming edge entering from the right – *East*; Fig. 10f). For the types R and L we further distinguish whether P passes to the left of N (R^{cc}/L^{cc}; Figs. 10c and 10d) or to the right of N (R^{c}/L^{c}; Figs. 10g and 10h): Let h be the horizontal line through N. We say that P *passes to the left (right)* of N if the last edge of P (from S to N) that intersects h does so to the left (right) of N. See Fig. 11. Thus, there are six *North-types* for a path between the poles: $R^{cc}, L^{cc}, W, E, R^{c}, L^{c}$. The superscripts c and cc stand for clockwise and counter-clockwise, respectively, to denote the rotation of a path that passes to the left (right) of N, when walking from N to S. This is justified in the next lemma and depicted e.g., in Fig. 12a, where the right S-N-path has type L^{c}, since (walking from N to S) it first bends to the left and then passes to the right of N by rotating clockwise.

Lemma 5 *Let G be a series-parallel DAG with no internal sources. Let an upward-planar L-drawing of G be given where the poles S and N of G are incident to the outer face and S is below N. Let P be a not necessarily directed S-N-path. Let P' be the polygonal chain obtained from P by adding a vertical segment pointing from N downward. The rotation of P' is*

- π *if the type of P at N is in $\{E, L^{c}, R^{c}\}$.*

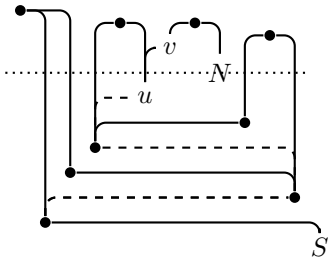


Figure 11: Solid edges indicate an S-N-path P of type L^{cc} at the North pole. Dashed edges are other edges of the graph. The path P passes to the left of N which is determined by the edge (u, v).

- *$-\pi$ if the type of P at N is in $\{L^{cc}, R^{cc}, W\}$.*

Proof: Observe that S is the bottommost vertex of G, since it is lower than N and no internal vertex of G is a source; refer to Fig. 13. Let h be the horizontal line through N and let h_ℓ (h_r) be the part of h to the left (right) of N. We claim that, since G has no internal source, the drawing of P does not contain a subcurve $\mathcal{C}$ in the half plane above h that connects h_ℓ and h_r: Otherwise, let e_ℓ and e_r, respectively, be the edges of P containing the two end points of $\mathcal{C}$. Let v_ℓ and v_r be the tails of e_ℓ and e_r, respectively. Since G has no internal source, there must be a descending path P_ℓ and P_r, respectively, from v_ℓ and v_r to a pole. Since the North pole is above v_ℓ and v_r, the descending paths must end at the South pole. However, this implies that the union of P, P_ℓ, and P_r contains a cycle in G enclosing N, contradicting that N is incident to the outer face.

We now first consider the case in which the type of P at N is any in $\{L^c, R^c, E\}$. Let $p = (x, y)$ be the end point of P' different from S, i.e., the point below N. We may assume that p is very close to N. The above claim implies that by shortening some vertical and horizontal parts of P', we can ensure that P' does not traverse the horizontal line through p to the left of p. This does not change the rotation of P'. Let $x_{\min}$ be the minimum x-coordinate of P'. Now consider the orthogonal polyline $Q : p, (x_{\min} - 1, y), (x_{\min} - 1, y_S), S$ where y_S is the y-coordinate of S. Concatenating P' and Q yields a simple polygon traversed in counterclockwise order. Thus the rotation of P equals the rotation of a polygon minus the rotation of the two convex bends in Q minus the convex bend in S minus the concave bend at p. Thus the rotation of P' is $2\pi - 3 \cdot \pi/2 + \pi/2 = \pi$.

If the type of P at N is in $\{L^{cc}, R^{cc}, W\}$ then we obtain the respective result, by concatenating the reversion of P' and the polygonal chain $S, (x_{\max} + 1, y_S), (x_{\max} + 1, y), p$ where $x_{\max}$ is the maximum x coordinate of P. Thus, the reversion of P' has rotation $2\pi - 3 \cdot \pi/2 + \pi/2 = \pi$ and the rotation of P' is the negative of it. □

(a) $\langle (E, L), (L^c, R) \rangle$ (b) $\langle (E, L), (L^c, L) \rangle$ (c) $\langle (R^{cc}, L), (R^c, R) \rangle$ (d) flag at N

Figure 12: The type of a component consists of the North- and South-type of the leftmost, and rightmost path P_L and P_R, respectively, as well as four flags indicating whether the first and the last bend on P_L and P_R is free, i.e. not contained in another edge.

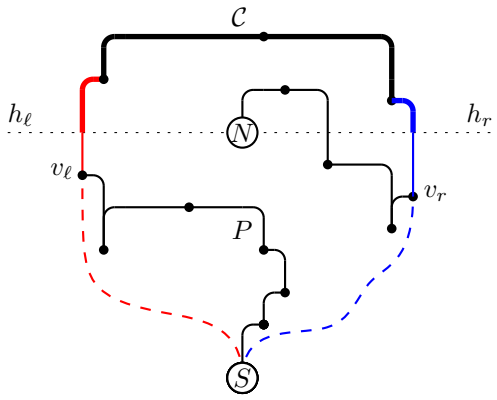


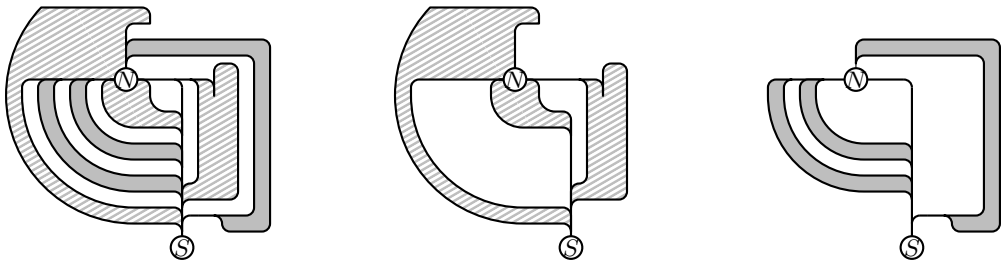
Figure 13: Illustration, supporting Section 5, showing that if a curve $\mathcal{C}$ (shown thicker) connects h_ℓ and h_r in the half-plane above N, then N is not incident to the outer face.

We say that the *type of a path between the poles* is (X, x), if X is the North-type and x is the South-type of the path, e.g., the type of a path that bends right at the South-pole, passes to the right of the North-pole and ends in an edge that leaves the North-pole bending to the left is (L^c, R), see P_R in Fig. 12a. For two North-types X and Y, we say $X < Y$ if X is before Y in the ordering $[R^{cc}L^{cc}WER^cL^c]$. The South-types are ordered $L < R$. For two types (X, x) and (Y, y) we say that $(X, x) \leq (Y, y)$ if $X \leq Y$ and $x \leq y$, and $(X, x) < (Y, y)$ if $(X, x) \leq (Y, y)$ and $X < Y$ or $x < y$. These orderings correspond to orderings of paths of the respective type from left to right at the South pole.

The *type of a component* is determined by eight entries, whether the component is a single edge or not, the choice of the bottommost pole (South pole), the type of P_L, the type of P_R, and additionally four FREE-*flags*: For each pole, two flags *left-free* and *right-free* indicating whether the bend on P_L and P_R, respectively, on the edge incident to the pole is *free* on the left or the right, respectively: More precisely, let P be an S-N-path and let e be an edge of P incident to a pole Z. We say that e is *free* on the right (left) at Z if e bends to the right (left) – walking from S to N – or if the bend on e is not contained in an edge not incident to Z; see Fig. 12d. We denote a type by $\langle (X, x), (Y, y) \rangle$ where (X, x) is the type of P_L and (Y, y) is the type of P_R without explicitly mentioning the flags or the choice of the South pole. Observe that $Y < L^c$ if $X = R^{cc}$ and $\langle (X, x), (Y, y) \rangle$ is the type of a component. Fig. 14a illustrates how components of different types can be composed in parallel.

Lemma 6 (Parallel Composition) *In the context of a canonical decomposition tree, a component C of type $\langle (X, x), (Y, y) \rangle$ with the given four* FREE*-flags can be obtained as a parallel composition of components $C_1, \ldots, C_\ell$ of type $\langle (X_1, x_1), (Y_1, y_1) \rangle, \ldots, \langle (X_\ell, x_\ell), (Y_\ell, y_\ell) \rangle$ with the respective* FREE*-flags from left to right at the South pole if and only if*

1. *$X_1 = X$, $Y_\ell = Y$, $x_1 = x$, $y_\ell = y$,*
2. *C is left(right)-free at the North- and South-pole, respectively, if and only if C_1 (C_ℓ) is,*
3. *At the South-pole*
   - *$y_i = x_{i+1} = L$ and C_i is right-free, or*



(a) parallel composition (b) partitioned into A (c) and B

Figure 14: (a) A parallel composition of eight components of the following types: $\langle (R^{cc}, L), (W, L)\rangle$, $\langle (W, L), (W, L)\rangle$, $\langle (W, L), (W, L)\rangle$, $\langle (W, L), (E, L)\rangle$, $\langle (E, L), (E, L)\rangle$ single edge, $\langle (E, R), (E, R)\rangle$ not left-free at N, $\langle (R^c, R), (R^c, R)\rangle$. The result is of type $\langle (R^{cc}, L), (R^c, R)\rangle$. (b+c) shows a partition of the set of components into two sets A and B.

- *$y_i = x_{i+1} = R$ and C_{i+1} is left-free, or*
- *$y_i = L$ and $x_{i+1} = R$ and C_i is right-free or C_{i+1} is left-free.*

4. *$Y_i \leq X_{i+1}$ and at the North-pole*
   - *C_i is right-free if $Y_i = X_{i+1} \in \{R^{cc}, E, R^c\}$*
   - *C_{i+1} is left-free if $Y_i = X_{i+1} \in \{L^{cc}, W, L^c\}$*
   - *C_i is right-free or C_{i+1} is left-free if $Y_i \in \{L^{cc}, L^c\}$ and $X_{i+1} \in \{R^{cc}, R^c\}$ or vice versa.*
5. *and single edges are the first among the components with a boundary path of type (W, R) and the last among the components with a boundary path of type (E, L).*

Proof: Necessity: Item 2 and Item 1 are obvious. We next prove Item 3. Assume that an upward-planar L-drawing of C is given. Let S and N be the North- and South-pole of C, respectively. In an upward-planar L-drawing the edges incident to S from left to right must be first all edges bending to the left and then all edges bending to the right. This yields $y_i \leq x_{i+1}$, $i = 1, \ldots, \ell - 1$.

Assume now that the rightmost S-N-path P_R of C_i has South-type L and that the bend b of the first edge e of P_R is contained in the horizontal segment of another edge e' of C_i. Then the first edge e'' of the leftmost path of C_{i+1} cannot bend to the left. Otherwise, e'' would have to intersect e or e'. Analogously, e cannot bend to the right if the leftmost path of C_{i+1} has South-type R and the bend on e'' is contained in another edge of C_{i+1}. Finally consider the case that e bends to the left and e'' bends to the right. If the vertical segment of e is longer than the vertical segment of e'', then the bend of e'' cannot be contained in the horizontal segment of another edge and vice versa.

Now, we handle Item 4. Any set of disjoint S-N-paths in the order from left to right at the South pole must have the North type in this order: $R^{cc}, L^{cc}, W, E, R^c, L^c$ where there cannot be both, paths of type L^c and paths of type R^{cc}. It follows that $Y_i \leq X_{i+1}$, $i = 1, \ldots, \ell - 1$. If the leftmost S-N-path P_L of a component C_i is of type R^{cc} (E) and if the bend b in the last edge e of P_L is not free at the North-pole, i.e., if there is an edge e' in C_i that is not incident to N but contains b, then the rightmost S-N-path of C_{i+1} cannot be of type R^{cc} (E). Similarly, if the rightmost S-N-path P_R of a component C_{i+1} is of type L^c (W) and if the bend b in the last edge e of P_R is not free at the North-pole, i.e., if there is an edge e' in C_{i+1} that is not incident to N but contains b, then the leftmost S-N-path of C_i cannot be of type L^c (W). Moreover, since C has

no source other than the poles, the rightmost path of type R^c and a leftmost path of type L^{cc} is always free. Finally, if the last edges of P_R and P_L bend to opposite sides, then the bend on the edge with the shorter vertical segment cannot be contained in the horizontal segment of another edge.

Finally, we discuss Item 5. Assume now that there is an edge (S, N) of North type W. Let C_i be the component preceding (S, N) and assume that the North-type of the rightmost path P_R of C_i is of type W. Then the South type of P_R cannot be R. Similarly, if (S, N) has North type E and the leftmost path P_L of the component succeeding (S, N) has also North type E, then the South type of P_L must be R.

Sufficiency: By construction, we ensure that the angle between two incoming edges is 0 or π and the angle between an incoming and an outgoing edge is $\pi/2$ or $3\pi/2$. It remains to show the following three conditions [11]: (i) The sum of the angles at a vertex is 2π, (ii) the rotation at any inner face is 2π, (iii) and the *bend-or-end property* is fulfilled, i.e., there is an assignment that assigns each edge to one of its end vertices with the following property. Let e_1 and e_2 be two incident edges that are consecutive in the cyclic order and attached to the same side of the common end vertex v. Let f be the face/angle between e_1 and e_2. Then at least one among e_1 and e_2 is assigned to v and its bend leaves a concave angle in f.

We first show that the bend-or-end property is fulfilled. To this end, we have to define an assignment of edges to end vertices: Consider upward-planar L-drawings of the components respecting the given types. Consider an edge e whose bend is contained in another edge e'. Let v be the common end vertex of e and e'. Assume first that e and e' are in the same component. Then we assign the bend on e to v.

Assume now that e and e' belong to different components. Then these components must be consecutive, say C_i and C_{i+1}. Moreover, e and e' are both the first or both the last edges of the rightmost S-N-path P_R of C_i and the leftmost S-N-path P_L of C_{i+1} or vice versa. In the following case distinction, the edge that is assigned is always the edge e, i.e., the edge whose bend is contained in the other.

We first consider the case in which e and e' are the first edges of P_L and P_R, i.e., e and e' are both incident to the South pole. If P_R and P_L both have South-type L, then e is the first edge of P_R. By Item 3, C_i must be right-free. Thus, we can assign the bend on e to S without violating any previous assignments. Similarly, if P_R and P_L both have South-type R, then we can assign the first edge of P_L to S without violating any previous assignments. If P_R has South type L and P_L has South type R, then we assign the first edge of P_R or P_L to S depending on whether C_i is right-free or C_{i+1} is left-free at the South pole. In Fig. 15, the assigned edge e is the first edge of P_L since C_{i+1} is left-free.

Now, we consider the case where e and e' are the last edges of P_L and P_R, i.e., e and e' are both incident to the North pole. We denote the last edges of P_R and P_L by e_R and e_L, respectively. If e and e' are both outgoing edges of N, the assignment is analogous to the South pole: If P_R and P_L have both North-type L^c or both L^{cc}, we assign the last edge of P_L to N. If P_R and P_L have both North-type R^c or both R^{cc}, we assign the last edge of P_R to N. If P_R has North type L^c or L^{cc} and P_L has North type R^c or R^{cc}, or vice versa, then we assign the last edge of P_R or P_L to N depending on whether C_i is right-free or C_{i+1} is left-free at the North pole.

Consider now the case that P_R and P_L are both of type W. Then C_{i+1} is left-free. Moreover, if C_{i+1} is a single edge, then C_i cannot have South-type R, thus, we have not assigned e_L to S. Hence, we can assign e_L to N. Finally, if P_R and P_L are both of type E we can assign e_R to N. We assign all edges that have not been assigned yet to an arbitrary end vertex. This assignment fulfils the bend-or-end property.

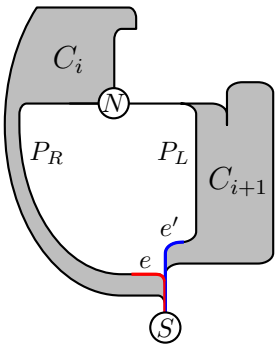


Figure 15: Assignment of the edge e when P_R has South type L and P_L has South type R, for the case in which C_{i+1} is left-free at the South pole.

By the construction, the angular sum around a vertex is always 2π. It remains to show that the rotation of all inner faces is 2π, which implies that the rotation of the outer face is -2π. If f is an inner face of a component, then its rotation is 2π by induction. So consider the face f_i between the components C_i and C_{i+1}. The boundary of f_i in counter-clockwise direction is combined by the reversed rightmost path P_R of C_i and the leftmost path P_L of C_{i+1}. Let P'_R be the concatenation of P_R and a short vertical segment s emanating under N and let P'_L be the concatenation of P_L and s. Consider the face f'_i bounded by P'_R and the reversion of P'_L. Then f_i has the same rotation as f'_i. Let $\mathrm{rot}(P)$ be the rotation of a path. The rotation of f'_i at S is $2 \cdot \pi/2$. The rotation of f'_i at the bottommost end point p of s is $2 \cdot (-\pi/2)$ or $2 \cdot \pi/2$, depending on whether the 2π angle between P'_R and P'_L at p is in the interior of f'_i or not, i.e. whether P_L and P_R pass to different sides of N or not. Thus, $\mathrm{rot}(f_i) = \mathrm{rot}(P'_R) \pm \pi - \mathrm{rot}(P'_L) + \pi$. Observe that by Section 5, $\mathrm{rot}(P'_R)$,$\mathrm{rot}(P'_L) \in \{\pm\pi\}$ on one hand and $\mathrm{rot}(P'_R) = -\mathrm{rot}(P'_L)$ if and only if P_L and P_R pass to different sides of N on the other hand. Thus, if they pass to the same side then $\mathrm{rot}(P'_R)$ and $-\mathrm{rot}(P'_L)$ cancel out and $\mathrm{rot}(f_i) = \mathrm{rot}(P'_R) + \pi - \mathrm{rot}(P'_L) + \pi = 2\pi$. In the other case $\mathrm{rot}(P'_R)$ and $-\mathrm{rot}(P'_L)$ sum to 2π and $\mathrm{rot}(f_i) = \mathrm{rot}(P'_R) - \pi - \mathrm{rot}(P'_L) + \pi = 2\pi$. In any case, it follows that the rotation of f_i is 2π. □

Section 5 yields a strict order of the possible types from left to right that can be composed in parallel. Moreover, let σ be a sequence of types of components from left to right that can be composed in parallel and let τ be a type in σ. Then Section 5 implies that τ appears exactly once in σ or the leftmost path and the rightmost path have both the same type in τ and all four free-flags are positive. In that case the type τ might occur arbitrarily many times and all appearances are consecutive. Thus, σ can be expressed as a *simple regular expression* on an alphabet $\mathcal{T}$, i.e., a sequence ρ of elements in $\mathcal{T} \cup \{^\star\}$ such that * does not occur as the first symbol of ρ and there are no two consecutive * in ρ. A sequence s of elements in $\mathcal{T}$ is *represented* by a simple regular expression ρ if it can be obtained from ρ by either removing the symbol preceding a * or by repeating it arbitrarily many times.

Observe that the elements from $\mathcal{T}$ in the simple regular expression ρ representing σ appear at most twice, once without a star and once with a star. Thus, the length of ρ is bounded by the number of types, i.e., constant. In particular, this means that there are only a constant number of expressions ρ which we need to consider in order to capture all possible feasible sequences σ.

In order to compute the type of a component, we proceed bottom-up on the binary decomposition tree. Observe that if a P-node has one or two P-node children then it does not suffice

(a) $\langle (R^{cc}, R), (R^{cc}, R) \rangle$ (b) $\langle (R^{c}, R), (R^{c}, R) \rangle$

Figure 16: An easy case of the series composition.

to put the drawings of the two sub components side by side. We have to merge the two subcomponents in order to consider all possible embeddings. See Fig. 14. Also recall that in the binary decomposition tree the components of a P-node μ are the subgraphs associated with the canonical descendants, i.e., the descendants μ' of μ that are S- or Q-nodes with the property that there are only P-nodes between μ and any μ'. The possible regular expressions associated with μ describe the ordering of the possible types of these components. Further, we associate with an S- or Q-node the set of regular expressions of length one that correspond to the different types of the respective components.

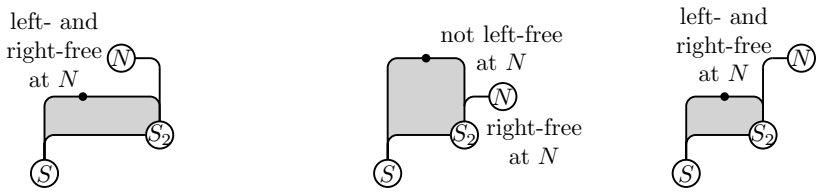
Lemma 7 *Let μ be a P-node of a binary decomposition tree T, let ν be the parent of μ, and let ν_A and ν_B be the children of μ. We can compute the collection of possible regular expressions and, thus, of all types of $G_\nu(\mu)$, in constant time under the condition that the possible regular expressions for $G_\mu(\nu_A)$ and $G_\mu(\nu_B)$ are known.*

Proof: Consider a planar L-drawing of $G_\nu(\mu)$. Let $\mu'_1, \ldots, \mu'_k$ be the canonical descendants of μ. Let σ be the sequence of types of the respective child components $G(\mu'_1), \ldots, G(\mu'_k)$ and let ρ be the respective regular expression. Consider the partition of $\mu'_1, \ldots, \mu'_k$ into the set A of canonical descendants of ν_A and the set B of canonical descendants of ν_B.

Notice that, by taking the components $G(\mu')$, $\mu' \in A$ in the order they appear in σ, we obtain a sequence σ_A which leads to a valid L-drawing of the corresponding components. This similarly holds for B. From this we can infer the following strategy to compute all possible regular expressions admitted by $G_\nu(\mu)$: We take any possible expression ρ for $G_\nu(\mu)$. Take a subsequence ρ_A of ρ of not necessarily consecutive symbols. Let ρ_B be the subsequence of ρ after removing the non-stared entries of ρ_A. If $G_\mu(\nu_A)$ and $G_\mu(\nu_B)$ realize ρ_A and ρ_B, respectively, then $G_\nu(\mu)$ realizes ρ. If no such split leads to ρ_A and ρ_B realizing $G_\mu(\nu_A)$ and $G_\mu(\nu_B)$ then $G_\nu(\mu)$ does not realize ρ. □

To understand how the type of a series composition is determined from the types of the children, let us first consider an easy example (see Figs. 16a and 16b): Assume that G_1 and G_2 consist both of a single edge e_1 and e_2, respectively, and that the type of both is (W, R). Assume further that G is obtained by merging the North poles N_1 and N_2 of G_1 and G_2, respectively. There are two ways how this can be done, namely e_1 can be attached to $N_1 = N_2$ before e_2 or after it in the counterclockwise order starting from R^{cc} and ending at L^c. In the first case the North type of G is R^{cc}, in the second case it is R^c. Moreover, in the first case G is left-free but not right-free at the North-pole, while in the second case it is right-free but not left-free at the South-pole.

Lemma 8 (Series Composition) *Let G_1 and G_2 be two series-parallel DAGs with no internal source that admit an upward-planar L-drawing of type $\langle (X_1, x_1), (Y_1, y_1) \rangle$ and $\langle (X_2, x_2), (Y_2, y_2) \rangle$, respectively, with the poles on the outer face. Let G be the DAG obtained by a series combination of G_1 and G_2, such that the common pole of G_1 and G_2 is not a source in at least one of G_1*



(a) G_1 has type L^{cc}, G_2 type L (b) G_1 has type L^{cc}, G_2 type R at S_2

Figure 17: Different free-flags in the case that the North pole is merged with the South pole

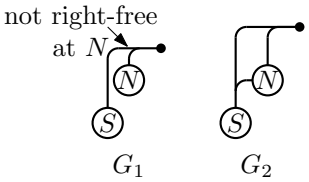


Figure 18: A series composition of G_1 and G_2 at the common North-pole is not possible with the given types since G_1 is not right-free at its North pole.

and G_2. Then the possible types of G in an upward-planar L-drawing maintaining the types of G_1 and G_2 can be determined in constant time.

Proof: Let S_i and N_i, respectively, be the South and North pole of G_i, $i = 1, 2$. We may assume that S_1 is the South pole of G and, thus, N_1 is the common pole of G_1 and G_2. We distinguish two cases, based on whether $N_1 = S_2$ or $N_1 = N_2$.

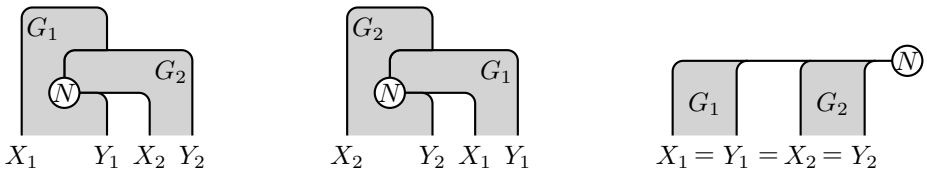
First suppose that $N_1 = S_2$, i.e., that N_2 is the North pole of G. It follows that on one hand N_1 cannot be a source of G_1. On the other G_2 must be contained in the external face of G_1. Thus, G admits an upward-planar L-drawing if and only if

- $x_2 = L \Rightarrow X_1 \notin \{R^{cc}, R^c\}$ and
- $y_2 = R \Rightarrow Y_1 \notin \{L^{cc}, L^c\}$.

The South-type of G is the South type of G_1. The North-type of G is the North-type of G_2 except for the FREE-flags, which might have to be updated if the next-to-last edge on the leftmost or rightmost path, respectively, is already contained in G_1 and is an outgoing edge of N_1. This might yield different North-types concerning the flags. See Fig. 17.

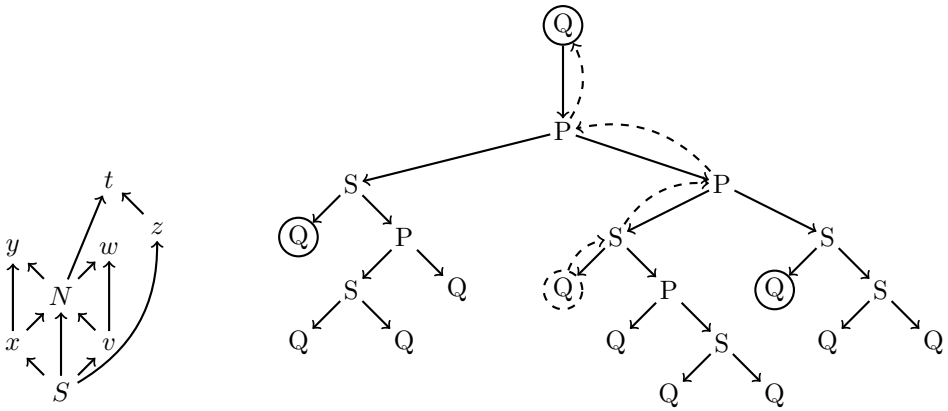
Now suppose that $N_1 = N_2$. Then G admits an upward-planar L-drawing if and only if one of the following three cases holds (see Fig. 19)

1. $X_1 < Y_2$ and $Y_1 \leq X_2$, and not ($X_1 = R^{cc}$ and $Y_2 = L^c$) implying that G_1 can only be embedded before G_2 or
2. $X_2 < Y_1$ and $Y_2 \leq X_1$, and not ($X_2 = R^{cc}$ and $Y_1 = L^c$) implying that G_2 can only be embedded before G_1, or
3. $X_1 = Y_1 = X_2 = Y_2 \in \{E, W\}$ which might give rise to two upward-planar L-drawings with distinct labels by adding G_2 before or after G_1 at the common pole. (Observe that $X_1 = Y_1 = X_2 = Y_2$ but not in $\{E, W\}$ is impossible since N_1 and N_2 are not both sources.)



(a) G_1 before G_2 (b) G_2 before G_1 (c) all North types equal

Figure 19: The three cases of a series composition at the North pole. In the third case G_1 can be before G_2 or vice versa.



(a) DAG G (b) Binary decomposition tree

Figure 20: How to find a suitable root of the decomposition tree in linear time.

and the respective FREE-conditions (see Fig. 18) are fulfilled at $N_1 = N_2$: namely assume that G_1 is embedded before G_2. In the case that the right path of G_1 uses the same port and bends to the same side as the left path of G_2 then G_1 has to be right-free and G_2 has to be left-free at the North pole. The necessity and sufficiency of these conditions can be proven analogously as in the parallel case.

The FREE-flags might have to be updated if the second edge on the leftmost or rightmost path, respectively, is already contained in G_2 or if the next-to-last edge on the leftmost or rightmost path, respectively, is already contained in G_1 and the type of G_1 at N_1 equals the respective type of G_2 at N_2. Except for the flags, the South-type of G is the South type of G_1 and the North type of G_2 yields the North type of G except for the specifications c or cc (also see Fig. 16): First observe that both, the leftmost path and the rightmost path, either have both type c or both type cc. Otherwise, G_2 would be contained in an inner face of G_1. The North type of both paths is indexed c if G_2 is embedded before G_1. Otherwise, the North type is indexed cc. Regarding the time complexity, observe that our computation of the set of possible types of G does not depend on the size of G_1 and G_2, but only on the number of types in their admissible sets. Since these sets have constant size and the above conditions on the types of G_1 and G_2 can be tested in constant time, we thus output the desired set in constant time. □

Finally, we discuss how to consider different rootings. Recall that we root the decomposition tree T at a Q-node corresponding to an edge incident to the single source s. Consider any tree

traversal from any suitable root. Label an edge with the direction parent to child. If the edge was already labeled in the same direction in a previous traversal, we stop the recursion, since at this point the type of the respective component was already determined. Bottom-up, we can now determine for each node the possible types and in addition for each P-node the possible regular expressions as described in Section 5 in constant time per node. Recall that the root corresponds to a final parallel composition and is treated as such. Over all choices of the root, we have to traverse each edge of the binary decomposition tree once in each direction. Thus, the whole run time is linear.

For an example see Fig. 20. The encircled Q-nodes represent edges incident to the single source S. We start with the topmost Q-node as a root. When choosing the dashed encircled Q-node as a root next, we only have to traverse the dashed edges in the decomposition tree. The types of all other subcomponents have already been determined.

Single Sink. For the case that G has a single sink, the algorithmic principles are the same as in the single-source case. The main difference is the type of an N-S-path P in a component C, where S and N are the South- and North-pole of C. The North pole of a component is always a sink and the North-type of P is W or E in this order from left to right. The South-type is one among $E^c, W^c, L, R, E^{cc}, W^{cc}$ in this order from left to right (according to the outgoing edges at N), depending on whether the last edge of P (traversed from N to S) is an incoming edge entering from the left (W) or the right (E), or an outgoing edge bending to the left (L) or the right (R), and whether the last edge of P leaving the half-space above the horizontal line through S does so to the right of S (cc) or the left of S (c).

The type consists again of the choice of the topmost pole (North pole), the type of the leftmost N-S-path, the type of the rightmost N-S-path, the four FREE-flags – which are defined the same way as in the single source case – and whether the component is a single edge or not.

6 Conclusion and Future Work

We have shown that a DAG admits an upward-planar L-drawing if and only if it is a subgraph of a planar st-graph that admits a bitonic st-ordering. This implies that DAGs admitting an upward-planar L-drawing also admit a straight-line upward-planar drawing in quadratic area. We have devised an algorithm to decide in linear time whether a plane single-source or -sink DAG admits an upward-planar L-drawing. A natural extension of this work would be to consider plane DAGs with multiple sinks and sources, the complexity of which is open. In the variable embedding setting, we have presented a linear-time testing algorithm for single-source or -sink series-parallel DAGs. Some next directions are to consider general single-source or -sink DAGs or general series-parallel DAGs. In the case of single-sink DAGs we could even handle the case in which all biconnected components are series-parallel in linear time. It would be interesting to study whether this result could be extended to the single-source case and, more generally, whether the single-source case is more complex than the single-sink case. We remark that the complexity of testing for the existence of upward-planar L-drawings of upward-planar graphs in general also remains open.

References

[1] P. Angelini, M. A. Bekos, H. Förster, and M. Gronemann. Bitonic st-orderings for upward planar graphs: Splits and bends in the variable embedding scenario. *Algorithmica*, 85(9):2667–2692, 2023. doi:10.1007/S00453-023-01111-5.

[2] P. Angelini, S. Chaplick, S. Cornelsen, and G. Da Lozzo. On upward-planar L-drawings of graphs. In S. Szeider, R. Ganian, and A. Silva, editors, *Mathematical Foundations of Computer Science (MFCS'22)*, volume 241 of *LIPIcs*, pages 10:1–10:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.MFCS.2022.10.

[3] P. Angelini, S. Chaplick, S. Cornelsen, and G. Da Lozzo. On upward-planar L-drawings of graphs. *CoRR*, abs/2205.05627, 2022. arXiv:2205.05627, doi:10.48550/arXiv.2205.05627.

[4] P. Angelini, S. Chaplick, S. Cornelsen, and G. Da Lozzo. Planar L-drawings of bimodal graphs. *Journal of Graph Algorithms and Applications*, 26(3):307–334, 2022. doi:10.7155/jgaa.00596.

[5] P. Angelini, G. Da Lozzo, M. Di Bartolomeo, V. Di Donato, M. Patrignani, V. Roselli, and I. G. Tollis. Algorithms and bounds for L-drawings of directed graphs. *Int. J. Found. Comput. Sci.*, 29(4):461–480, 2018. doi:10.1142/S0129054118410010.

[6] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(4):476–497, 1994. doi:10.1007/BF01188716.

[7] P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM Journal on Computing*, 27(1):132–169, 1998. doi:10.1137/S0097539794279626.

[8] C. Binucci, G. D. Lozzo, E. D. Giacomo, W. Didimo, T. Mchedlidze, and M. Patrignani. Upward book embeddability of st-graphs: Complexity and algorithms. *Algorithmica*, 85(12):3521–3571, 2023. doi:10.1007/S00453-023-01142-Y.

[9] G. Borradaile, P. N. Klein, S. Mozes, Y. Nussbaum, and C. Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM Journal on Computing*, 46(4):1280–1303, 2017. doi:10.1137/15M1042929.

[10] G. Brückner, M. Himmel, and I. Rutter. An SPQR-tree-like embedding representation for upward planarity. In D. Archambault and C. D. Toth, editors, *Graph Drawing and Network Visualization (GD'19)*, volume 11904 of *LNCS*, pages 517–531. Springer, 2016. doi:10.1007/978-3-030-35802-0_39.

[11] S. Chaplick, M. Chimani, S. Cornelsen, G. Da Lozzo, M. Nöllenburg, M. Patrignani, I. G. Tollis, and A. Wolff. Planar L-drawings of directed graphs. *Comput. Geom. Topol.*, 2(1):7:1–7:15, 2023. URL: https://www.cgt-journal.org/index.php/cgt/article/view/43.

[12] S. Chaplick, E. Di Giacomo, F. Frati, R. Ganian, C. N. Raftopoulou, and K. Simonov. Parameterized algorithms for upward planarity. In X. Goaoc and M. Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPIcs*, pages 26:1–26:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.SoCG.2022.26.

[13] S. Chaplick, E. D. Giacomo, F. Frati, R. Ganian, C. N. Raftopoulou, and K. Simonov. Testing upward planarity of partial 2-trees. In P. Angelini and R. von Hanxleden, editors, *Graph Drawing and Network Visualization - 30th International Symposium, GD 2022, Tokyo, Japan, September 13-16, 2022, Revised Selected Papers*, volume 13764 of *Lecture Notes in Computer Science*, pages 175–187. Springer, 2022. doi:10.1007/978-3-031-22203-0_13.

[14] S. Cornelsen and G. Diatzko. Planar confluent orthogonal drawings of 4-modal digraphs. *J. Graph Algorithms Appl.*, 27(7):523–540, 2023. doi:10.7155/JGAA.00632.

[15] G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theor. Comput. Sci.*, 61:175–198, 1988. doi:10.1016/0304-3975(88)90123-5.

[16] G. Di Battista, R. Tamassia, and I. G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discret. Comput. Geom.*, 7:381–401, 1992. doi:10.1007/BF02187850.

[17] W. Didimo, F. Giordano, and G. Liotta. Upward spirality and upward planarity testing. *SIAM Journal on Discrete Mathematics*, 23(4):1842–1899, 2010. doi:10.1137/070696854.

[18] F. Frati. On minimum area planar upward drawings of directed trees and other families of directed acyclic graphs. *International Journal of Computational Geometry & Applications*, 18(3):251–271, 2008. doi:10.1007/978-3-540-74839-7_13.

[19] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.

[20] M. Gronemann. Bitonic *st*-orderings for upward planar graphs. In Y. Hu and M. Nöllenburg, editors, *Graph Drawing and Network Visualization (GD'16)*, volume 9801 of *LNCS*, pages 222–235. Springer, 2016. Available at https://arxiv.org/abs/1608.08578.

[21] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In J. Marks, editor, *Graph Drawing, 8th International Symposium, GD 2000, Colonial Williamsburg, VA, USA, September 20-23, 2000, Proceedings*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2000. doi:10.1007/3-540-44541-2_8.

[22] M. D. Hutton and A. Lubiw. Upward planar drawing of single-source acyclic digraphs. *SIAM Journal on Computing*, 25(2):291–311, 1996. doi:10.1137/S0097539792235906.

[23] H. C. Purchase. Metrics for graph drawing aesthetics. *J. Vis. Lang. Comput.*, 13(5):501–516, 2002. doi:10.1006/jvlc.2002.0232.

[24] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(2):109–125, 1981. doi:10.1109/TSMC.1981.4308636.

[25] R. Tamassia. *Handbook of Graph Drawing and Visualization*. Chapman & Hall/CRC, 1st edition, 2016.

[26] C. Ware, H. C. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Inf. Vis.*, 1(2):103–110, 2002. doi:10.1057/palgrave.ivs.9500013.