# On Upward-Planar L-Drawings of Graphs

**Patrizio Angelini**
John Cabot University, Rome, Italy

**Steven Chaplick**
Maastricht University, The Netherlands

**Sabine Cornelsen**
University of Konstanz, Germany

**Giordano Da Lozzo**
Roma Tre University, Rome, Italy

## Abstract

In an *upward-planar L-drawing* of a directed acyclic graph (DAG) each edge $e$ is represented as a polyline composed of a vertical segment with its lowest endpoint at the tail of $e$ and of a horizontal segment ending at the head of $e$. Distinct edges may overlap, but 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. On the combinatorial side, 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 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: (i) when the drawing must respect a prescribed embedding and (ii) when no restriction is given on the embedding, but the DAG is biconnected and series-parallel.

**2012 ACM Subject Classification** Mathematics of computing → Graph theory; Mathematics of computing → Graphs and surfaces

**Keywords and phrases** graph drawing, planar L-drawings, directed graphs, bitonic st-ordering, upward planarity, series-parallel graphs

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2022.10

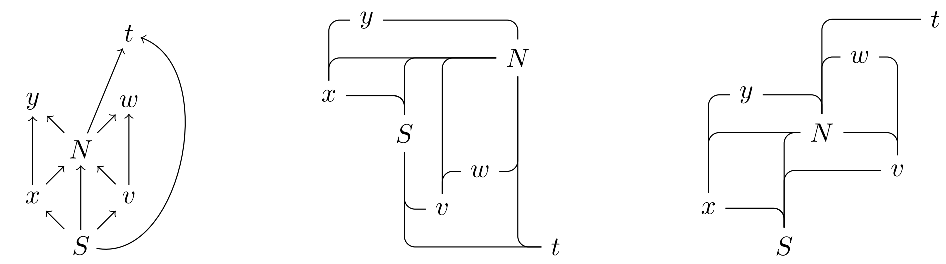
**Related Version** *Full Version*: https://arxiv.org/abs/2205.05627

**Funding** *Sabine Cornelsen*: Funded by the DFG – Project-ID 50974019 – TRR 161 (B06).
*Giordano Da Lozzo*: Supported in part by MIUR grant 20174LF3T8 *"AHeAD"*.

## 1 Introduction

In order to visualize hierarchies, directed acyclic graphs (DAGs) are often drawn in such 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. Sugiyama et al. [21] provided a general framework for drawing DAGs upward. To support readability, it is desirable to avoid edge crossings [20, 23]. 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 [12] showed that a


47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022).
Editors: Stefan Szeider, Robert Ganian, and Alexandra Silva; Article No. 10; pp. 10:1–10:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany




**(a)** DAG $G$. **(b)** Planar L-drawing. **(c)** Upward-planar L-drawing of $G - \{S, t\}$.

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

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 [5, 8]. However, it is NP-complete to decide whether a DAG admits an upward-planar drawing when no fixed embedding is given [16]. 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 [6, 9, 19], or if the underlying undirected graph is series-parallel [14]. Furthermore, parameterized algorithms for upward-planarity testing exist with respect to the number of sources or the treewidth of the input DAG [11].

Every upward-planar DAG admits a straight-line upward-planar drawing [12], however such a drawing may require exponential area [13]. Gronemann introduced bitonic st-orderings for DAGs [17]. A plane st-graph that admits a bitonic st-ordering also admits an upward-planar drawing in quadratic area. It can be tested in linear time whether a plane st-graph admits a bitonic st-ordering [17], and whether a planar st-graph admits a planar embedding that allows for a bitonic st-ordering [1, 10]. 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 the fixed embedding setting [17].

In an *L-drawing* of a directed graph [4] 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 NP-complete [10]. On the other hand, every upward-planar DAG admits a planar L-drawing [3]. A planar L-drawing is *upward* if the lowest end vertex of the vertical segment of an edge $e$ is the tail of $e$ (see Fig. 1c). A planar st-graph admits an upward-planar L-drawing if and only it admits a bitonic st-ordering [10].

**Our Contribution.** We characterize in Theorem 2 the plane DAGs admitting an upward-planar L-drawing as the subgraphs of plane st-graphs admitting a bitonic st-ordering. 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 (Theorem 3). Moreover, the characterization allows 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 (Theorem 5).

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 (Theorem 4). The respective result holds for single-sink acyclic cacti. Finally, we use a dynamic-programming approach combined with a matching algorithm for regular expressions to decide in linear time whether a DAG with a single source or a single sink has an embedding admitting an upward-planar L-drawing if it is biconnected and series-parallel (Theorem 7). Observe that a plane st-graph does not necessarily admit an upward-planar L-drawing if the respective graph with reversed edges does. This justifies studying single-source and -sink graphs independently. Full details for proofs of statements marked with ($\star$) can be found in the full version of the paper [2].

## 2 Preliminaries

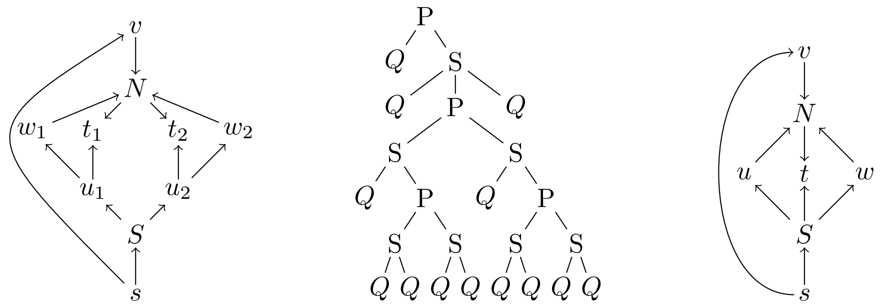
For standard graph theoretic notations and definitions we refer the reader to [22].

**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.

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 $2\pi$ angle to the left, we handle this as two concave angles and if there is a 0 angle to the left, we handle this as two convex angles. The rotation of a simple polygon – with possible overlaps of consecutive edges – traversed in counterclockwise direction is $2\pi$.

**Single-source series-parallel DAGs.** Series-parallel digraphs are digraphs with two distinguished vertices, called *poles*, and can be defined recursively as follows: A single edge is a series-parallel digraph. Given $k$ series-parallel digraphs $G_1, \ldots, G_k$ (*components*), with poles $v_i, u_i$, $i = 1, \ldots, k$, a series-parallel digraph $G$ with poles $v$ and $u$ can be obtained in two




**(a)** DAG not upward-planar. **(b)** Decomposition-tree. **(c)** Embedding not upward-planar.

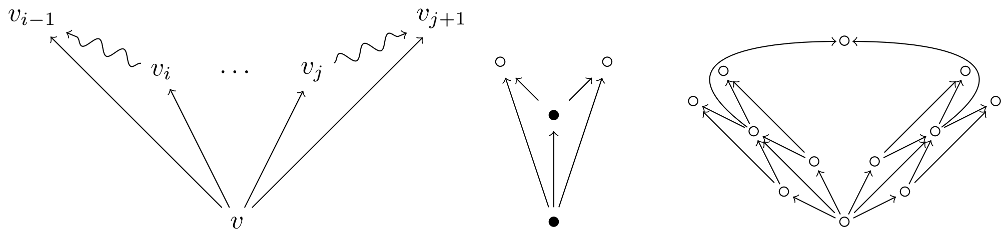
**Figure 2** a) A bimodal single-source series-parallel DAG that is not upward-planar b) with its decomposition-tree. c) A single-source series-parallel DAG with an embedding that is not upward-planar. However, the DAG with a different embedding is upward-planar.

ways: by merging $v_1, \ldots, v_k$ and $u_1, \ldots, u_k$, respectively, into the new poles $v$ and $u$ (*parallel composition*), or by merging the vertices $u_i$ and $v_{i+1}$, $i = 1, \ldots, k-1$, and setting $u = u_1$ and $v = v_k$ (*series composition*). The recursive construction of a series-parallel digraph is represented in a decomposition-tree $T$. We refer to the vertices of $T$ as *nodes*. The leaves (vertices of degree one) of the decomposition-tree are labeled Q and represent the edges. The other nodes (*inner nodes*) are labeled $P$ for parallel composition or $S$ for series composition. No two adjacent nodes of $T$ have the same label. Fig. 2b shows the decomposition-tree of the graph in Fig. 2a. Let $\mu$ be a node of $T$. We denote by $T(\mu)$ the subtree rooted at $\mu$ and by $G(\mu)$ the subgraph of $G$ corresponding to $T(\mu)$, i.e., the subgraph of $G$ formed by the edges corresponding to the leaves of $T(\mu)$. The vertices of $G(\mu)$ that are different from its poles are called *internal*. 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 [18]. Moreover, rooting a decomposition-tree of a biconnected series-parallel digraph $G$ at an arbitrary inner node yields again a decomposition-tree of $G$.

In the following, we assume that $G$ is a series-parallel DAG with a single source (sink) $s$. If $G$ has more than one edge, we root the decomposition-tree $T$ at the inner node incident to the Q-node corresponding to an edge incident to $s$. This implies that for any node $\mu$ 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 [6] 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. 2c. 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 $\mu$ with two children $\mu_1$ and $\mu_2$ 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 $G(\mu_2)$. Bimodal single-source series-parallel DAGs without this property are always upward-planar [2].

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 the South (North) pole of $G$ is the unique source (sink) $s$. If $\mu$ 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 $\mu$ is an $S$-node with children $\mu_1, \ldots, \mu_\ell$, then observe




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

**Figure 3** (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.

that at most one among the components $G(\mu_i)$, $i = 1, \dots, \ell$ 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).

**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 $\pi$ 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 [12].

For each vertex $v$ of a plane st-graph, we consider the ordered list $S(v) = \langle v_1, v_2, \dots, v_k \rangle$ 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 $S(v) = \langle v_1, v_2, \dots, v_k \rangle$ such that $\pi(v_i) < \pi(v_{i+1})$, $i = 1, \dots, h-1$, and $\pi(v_i) > \pi(v_{i+1})$, $i = h, \dots, k-1$. We say that the successor list $S(v) = \langle v_1, v_2, \dots, v_k \rangle$ 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. 3. Gronemann [17] characterized the plane st-graphs that admit a bitonic st-ordering as follows.

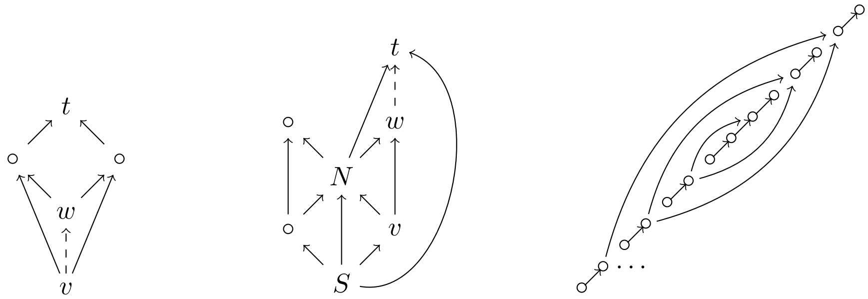
▶ **Theorem 1** ([17]). *A plane st-graph admits a bitonic st-ordering if and only if the successor list of no vertex contains a valley.*

## 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 [10]. We extend this result to general plane DAGs and discuss some consequences.

▶ **Theorem 2.** *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 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 top or the right, 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 $t$. ◀




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

**Figure 4** 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.

Observe that every series-parallel st-graph admits a bitonic st-ordering [1, 10] 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. 3b and 3c 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. 4a (without the dashed edge). $G$ has a unique upward-planar embedding. Since no vertex has more than two successors there cannot be a valley. Assume $G$ admits a planar L-drawing. By Theorem 2 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 Theorem 1, $G'$ is not bitonic, a contradiction.

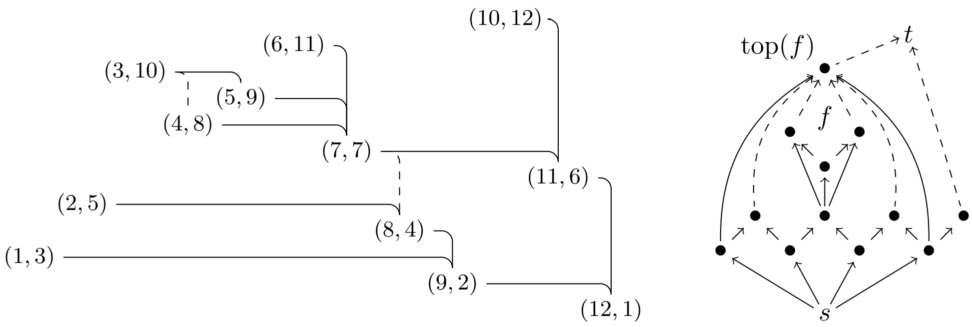
Now consider the DAG $G$ in Fig. 4b (without the dashed edge). $G$ has two symmetric upward-planar embeddings: with the curved edge to the right or the left of the remainder of the DAG. We may assume that the curved edge is to the right. But then an augmentation to a plane st-graph $G'$ must contain the dashed edge, which completes a valley at the single source and its three rightmost outgoing edges. By Theorem 1, $G'$ is not bitonic, a contradiction.

A planar L-drawing is *upward-leftward* [10] 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 [15, 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. 4c. 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 Theorem 2, 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 [10]. Hence, $G$ (and thus $T_k$) admits a straight-line upward-planar drawing in quadratic area $(2n - 2) \times (n - 1)$ [17], a contradiction. ◀




**(a)** Cactus with post- and pre-order numbers as coordinates. **(b)** Augmentation.

**Figure 5** (a) Single-source acyclic cactus. The dashed edges are the last edges on a left path cycles. (b) A new sink $t$ and the dashed edges augment a plane single-source DAG to a plane st-graph.

## 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.

▶ **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 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.** 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 $P_\ell$ and a *right* path $P_r$ between its single source and single sink. By flipping the cycle $C$ – maintaining outerplanarity – we can ensure that $P_\ell$ 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 $P_\ell$ 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 $P_\ell$, excluding $s_C$, a postorder number and then we continue backtracking on $P_r$. See Fig. 5a.

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, the drawing is upward-leftward. In [2] we prove that it is also planar and preserves the embedding, which was updated only in the presence of transitive edges.

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. ◀

**General DAGs.** 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 $\pi$ 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 [6].

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. [5] 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. An explicit proof for single-source or -sink DAGs can be found in [2].

▶ **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.*

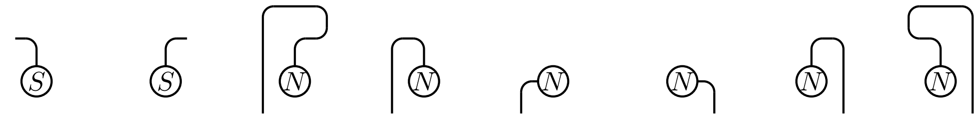
In the following, we prove the theorem for a DAG $G$ with a single source $s$; the single-sink case is discussed in [2]. In an upward-planar straight-line drawing of $G$, the only large angle at a source-switch is the angle at $s$ in the outer face. Thus, in the outer face all angles at sink-switches are large and in an inner face $f$ all but one angle at sink-switches are large. For an inner face $f$, let $\text{top}(f)$ be the sink-switch of $f$ without large angle. See Fig. 5b.

▶ **Lemma 6.** *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 of Theorem 5, 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 [17].

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 [10]. 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 Theorem 1.




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

**Figure 6** The different types of a path between the poles. (a,b) South types; (c-h) North types.

Namely, the edges to the new sink $t$ cannot be contained in any directed cycle. Furthermore, by Theorem 2, 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 Lemma 6, 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 Lemma 6, there would be such a path in any augmentation of $G$ to a planar st-graph. 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. ◀

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

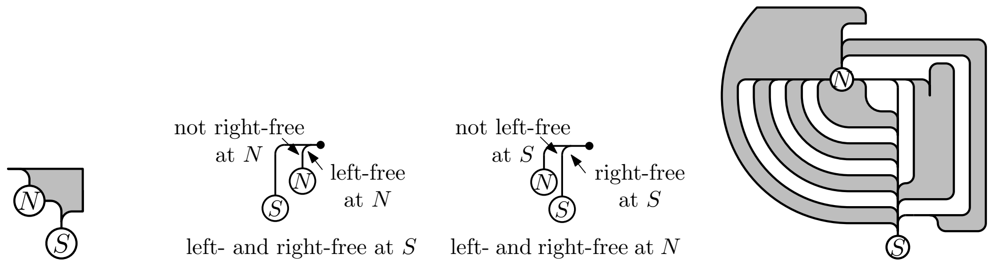
The goal of this section is to prove the following theorem.

▶ **Theorem 7.** *It can be tested in linear time whether a DAG with a single source or a single sink admits an upward-planar L-drawing if it is biconnected and series-parallel.*

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. [7] and Chaplick et al. [11]. 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 a parallel or series composition, 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. 6a) and $R$ (outgoing edge bending to the *right*; Fig. 6b). 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. 6e) and $E$ (incoming edge entering from the right – *East*; Fig. 6f). For the types $R$ and $L$ we further distinguish whether $P$ passes to the left of $N$ ($R^{cc}/L^{cc}$; Figs. 6c and 6d) or to the right of $N$ ($R^c/L^c$; Figs. 6g and 6h): Let $h$ be the horizontal line




**(a)** $\langle(E,L),(L^c,R)\rangle$ **(b)** $\langle(R^{cc},R),(R^{cc},R)\rangle$ **(c)** $\langle(R^c,R),(R^c,R)\rangle$ **(d)** Parallel composition.

**Figure 7** (a–c) Illustrations for types of a component. (d) 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 $\langle(R^{cc},L),(R^c,R)\rangle$.

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 $N$. 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. 7a, 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 8** (⋆). *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 $N$ 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*
- *$\pi$ if the type of $P$ at $N$ is in $\{E, L^c, R^c\}$.*
- *$-\pi$ if the type of $P$ at $N$ is in $\{L^{cc}, R^{cc}, W\}$.*

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. 7a. 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 $x < y$.

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 $X$. We say that $e$ is *free* on the right (left) at $X$ 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 $X$. See Figs. 7b and 7c. 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. 7d illustrates how components of different types can be composed in parallel.

▶ **Lemma 9** (Parallel Composition (⋆)). *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 $\langle(X_1, x_1), (Y_1, y_1)\rangle, \ldots, \langle(X_\ell, x_\ell), (Y_\ell, y_\ell)\rangle$ from left to right at the South pole if and only if*

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

**Sketch of Proof.** Since the necessity of the conditions is evident, we shortly sketch how to prove sufficiency. By construction, we ensure that the angle between two incoming edges is 0 or $\pi$ 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 [10]: (i) The sum of the angles at a vertex is $2\pi$, (ii) the rotation at any inner face is $2\pi$, (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 $f$. ◀

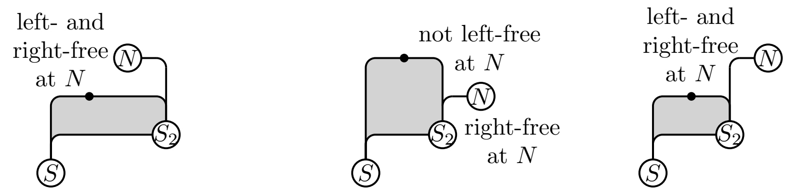
Lemma 9 yields a strict order of the possible types from left to right that can be composed in parallel. Moreover, let $\sigma$ be a sequence of types of components from left to right that can be composed in parallel and let $\tau$ be a type in $\sigma$. Then Lemma 9 implies that $\tau$ appears exactly once in $\sigma$ or the leftmost path and the rightmost path have both the same type in $\tau$ and all four free-flags are positive. In that case the type $\tau$ might occur arbitrarily many times and all appearances are consecutive. Thus, $\sigma$ can be expressed as a *simple regular expression* on an alphabet $\mathcal{T}$, i.e., a sequence $\rho$ of elements in $\mathcal{T} \cup \{^\star\}$ such that $^\star$ does not occur as the first symbol of $\rho$ and there are no two consecutive $^\star$ in $\rho$. A sequence $s$ of elements in $\mathcal{T}$ is *represented* by a simple regular expression $\rho$ if it can be obtained from $\rho$ by either removing the symbol preceding a $^\star$ or by repeating it arbitrarily many times.

Observe that the elements in the simple regular expression $\rho$ representing $\sigma$ are distinct, thus, the length of $\rho$ is linear in the number of types, i.e., constant. In particular, to obtain a linear-time algorithm to enumerate the attainable types of a series-parallel DAG obtained via a parallel composition, it suffices to establish the following algorithmic lemma.

▶ **Lemma 10** (Simple Regular Expression Matching ($\star$)). *Let $\mathcal{T}$ be a constant-size alphabet (set of types), and $\rho$ be a constant-length simple regular expression over $\mathcal{T}$. For a collection $\mathcal{C}$ of items where each $C \in \mathcal{C}$ has a set $\mathcal{T}(C) \subseteq \mathcal{T}$, one can test in $\mathcal{O}(|\mathcal{C}|)$ time, if there is a selection of a type from each $\mathcal{T}(C)$, $C \in \mathcal{C}$ that can be ordered to obtain a sequence represented by $\rho$.*

▶ **Corollary 11.** *The types of a parallel composition can be computed in time linear in the number of its children.*

In order to understand how the type of a series composition is determined from the types of the children, let us first have a look at an easy example: 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.




**(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 8** Different free-flags in the case that the North pole is merged with the South pole.

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. See Figs. 7b and 7c.

▶ **Lemma 12** (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 a certain 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 both, $G_1$ 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 $G_2$.

First suppose that $N_1 = S_2$, i.e., that $N_2$ is the North pole of $G$. It follows that $N_1$ cannot be a source of $G_1$. Then $G$ admits an upward-planar L-drawing if and only if $x_2 = L$ and $X_1 \neq R^{cc}$ or $y_2 = R$ and $Y_1 \neq L^c$, and the respective FREE-conditions are fulfilled at $N_1 = S_2$. 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. 8.

Now suppose that $N_1 = N_2$. Then $G$ admits an upward-planar L-drawing if and only if $G_1$ can be embedded before $G_2$, i.e., $Y_1 \leq X_2$ and $X_1 < Y_2$, and not ($X_1 = R^{cc}$ and $Y_2 = L^c$) or $G_2$ can be embedded before $G_1$, i.e., $Y_2 \leq X_1$ and $X_2 < Y_1$, and not ($X_2 = R^{cc}$ and $Y_1 = L^c$) and the respective FREE-conditions are fulfilled at $N_1 = N_2$. If $X_1 = Y_1 = X_2 = Y_2 \in \{E, W\}$ then both conditions are fulfilled 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. 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 $cc$: First observe that both, the leftmost path and the rightmost path, either have both type $c$ or both type $cc$. Otherwise, $G_1$ would be contained in an inner face of $G_2$. 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. ◀

▶ **Lemma 13.** *The types of a series composition can be computed in time linear in the number of its children.*

**Proof.** Let $C_1, \ldots, C_\ell$ be the components of a series component $C$ and let $\mathcal{T}(C_i)$, $i = 1, \ldots, \ell$ be the set of possible types of $C_i$. For $k = 1, \ldots, \ell$, we inductively compute the set $\mathcal{T}_i$ of possible types of the series combination $C^k$ of $C_1, \ldots, C_k$, where $\mathcal{T}_1 = \mathcal{T}(C_1)$. To compute $\mathcal{T}_k$ for some $k = 2, \ldots, \ell$, we combine all possible combinations of a type in $\mathcal{T}_{k-1}$ and a type in $\mathcal{T}(C_k)$ and, applying Lemma 12, we check in constant time which types (if any) they would yield for $C^k$. Since the number types is constant each step can be done in constant time. ◀

**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 the information whether the component is a single edge or not.

## 6 Conclusion and Future Work

We have shown how 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. We remark that the complexity of testing for the existence of upward-planar L-drawings in general also remains open.

## References


1. Patrizio Angelini, Michael A. Bekos, Henry Förster, and Martin Gronemann. Bitonic st-orderings for upward planar graphs: The variable embedding setting. In Isolde Adler and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science - 46th International Workshop, WG 2020*, volume 12301 of *LNCS*, pages 339–351. Springer, 2020. doi:10.1007/978-3-030-60440-0_27.
2. Patrizio Angelini, Steven Chaplick, Sabine Cornelsen, and Giordano Da Lozzo. On upward-planar L-drawings of graphs. *CoRR*, abs/2205.05627, 2022. doi:10.48550/arXiv.2205.05627.
3. Patrizio Angelini, Steven Chaplick, Sabine Cornelsen, and Giordano Da Lozzo. Planar L-drawings of bimodal graphs. *Journal of Graph Algorithms and Applications*, 26(3):307–334, 2022. doi:10.7155/jgaa.00596.
4. Patrizio Angelini, Giordano Da Lozzo, Marco Di Bartolomeo, Valentino Di Donato, Maurizio Patrignani, Vincenzo Roselli, and Ioannis 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.

5 Paola Bertolazzi, Giuseppe Di Battista, Giuseppe Liotta, and Carlo Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(4):476–497, 1994. doi:10.1007/BF01188716.

6 Paola Bertolazzi, Giuseppe Di Battista, Carlo Mannino, and Roberto Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM Journal on Computing*, 27(1):132–169, 1998. doi:10.1137/S0097539794279626.

7 Carla Binucci, Giordano Da Lozzo, Emilio Di Giacomo, Walter Didimo, Tamara Mchedlidze, and Maurizio Patrignani. Upward book embeddings of st-graphs. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019*, volume 129 of *LIPIcs*, pages 13:1–13:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.SoCG.2019.13.

8 Glencora Borradaile, Philip N. Klein, Shay Mozes, Yahav Nussbaum, and Christian 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.

9 Guido Brückner, Markus Himmel, and Ignaz 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.

10 Steven Chaplick, Markus Chimani, Sabine Cornelsen, Giordano Da Lozzo, Martin Nöllenburg, Maurizio Patrignani, Ioannis G. Tollis, and Alexander Wolff. Planar L-drawings of directed graphs. In Fabrizio Frati and Kwan-Liu Ma, editors, *GD 2017*, volume 10692 of *LNCS*, pages 465–478. Springer, 2018. doi:10.1007/978-3-319-73915-1_36.

11 Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov. Parameterized algorithms for upward planarity. In *38th International Symposium on Computational Geometry, SoCG 2022*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

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

13 Giuseppe Di Battista, Roberto Tamassia, and Ioannis G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discret. Comput. Geom.*, 7:381–401, 1992. doi:10.1007/BF02187850.

14 Walter Didimo, Francesco Giordano, and Giuseppe Liotta. Upward spirality and upward planarity testing. *SIAM Journal on Discrete Mathematics*, 23(4):1842–1899, 2010. doi:10.1137/070696854.

15 Fabrizio 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.

16 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.

17 Martin Gronemann. Bitonic *st*-orderings for upward planar graphs. In Yifan Hu and Martin Nöllenburg, editors, *Graph Drawing and Network Visualization (GD'16)*, volume 9801 of *LNCS*, pages 222–235. Springer, 2016. Available at arXiv:1608.08578.

18 Carsten Gutwenger and Petra Mutzel. A linear time implementation of SPQR-trees. In Joe 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.

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

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

21 Kozo Sugiyama, Shôjirô Tagawa, and Mitsuhiko 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.

**22** Roberto Tamassia. *Handbook of Graph Drawing and Visualization*. Chapman & Hall/CRC, 1st edition, 2016.

**23** Colin Ware, Helen C. Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Inf. Vis.*, 1(2):103–110, 2002. doi:10.1057/palgrave.ivs.9500013.