# Upward Planarity Testing*

*Ashim Garg* *Roberto Tamassia*

Department of Computer Science
Brown University
Providence, RI 02912–1910

**Abstract**

Acyclic digraphs, such as the covering digraphs of ordered sets, are usually drawn upward, i.e., with the edges monotonically increasing in the vertical direction. A digraph is upward planar if it admits an upward planar drawing. In this survey paper, we overview the literature on the problem of upward planarity testing. We present several characterizations of upward planarity and describe upward planarity testing algorithms for special classes of digraphs, such as embedded digraphs and single-source digraphs. We also sketch the proof of NP-completeness of upward planarity testing.

(February 1995)

*Research supported in part by the National Science Foundation under grant CCR–9423847. Email addresses of the authors: ag@cs.brown.edu, rt@cs.brown.edu.

# Contents

# 1 Introduction

Planarity is a fundamental property in graph visualization applications, and a substantial effort has been devoted to the design of efficient algorithms for testing planarity and for constructing planar drawings (see, e.g., [11]).

When considering acyclic digraphs, such as the covering digraphs of ordered sets, the notion of planarity needs to be refined to take into account the fact that such digraphs are usually drawn *upward*, i.e., with the edges monotonically increasing in the vertical direction. Namely, we say that a digraph is *upward planar* if it admits an upward planar drawing, and we say that an ordered set is planar if its covering digraph (Hasse diagram) is upward planar.

The study of upward planarity has been especially active in the last two decades, and has revealed fascinating connections with fundamental graph- and order-theoretic properties such as connectivity and dimension (see, e.g., [26]). By the end of 1993, upward planarity testing algorithms were known only for specific subclasses of digraphs, such as embedded and single-source digraphs, when the authors came up with a proof that the problem of testing whether a digraph is upward planar is NP-complete [17]. It is interesting to observe that while testing whether a graph admits a planar drawing or an upward drawing can be done in linear time, combining the two properties makes the problem NP-hard.

In this paper, we overview the literature on the problem of upward planarity testing. Basic definitions are given in Section 2. In Section 3, we present several characterizations of upward planarity, with special emphasis on those that yield efficient algorithms. Section 4 surveys upward planarity testing algorithms for special classes of digraphs, such as embedded digraphs and single-source digraphs. Finally, Section 5 gives a sketch of the NP-completeness proof of upward planarity testing.

Rigorous proofs and technical details are not provided in this paper. The interested reader can find them in the referenced literature. Theorems and lemmas without a reference can be considered "folklore".

# 2 Definitions and Preliminaries

We assume standard definitions on graphs, ordered sets, and algorithms. See, e.g., [6, 7, 9]. All the graphs and ordered sets in this paper are finite. We denote with $n$ the size of the graph or ordered set currently being considered.

An *embedding* of a planar graph is the collection of circular permutations of the edges incident upon each vertex in a planar drawing of the graph. Two drawings with the same embedding have the same faces, though they may have a different choice of the external face. An *embedded graph* is a planar graph equipped with an embedding. The *angles* of an embedded graph are the pairs of consecutive edges incident on the same vertex. Such angles are mapped to geometric angles in a straight-line drawing of the graph. A planar graph is said to be *outerplanar* it it has an embedding such that all the vertices are on the same face. The above definitions are also applicable to digraphs, by ignoring the direction of the edges.

A *source* of a digraph is a vertex without incoming edges. A *sink* of a digraph is a vertex without outgoing edges. An *internal vertex* of a digraph is a vertex that is not a source or a sink. An *acyclic* digraph is a digraph without directed cycles.

A drawing of a graph is *straight-line* if every edge is drawn as a straight-line segment. A

drawing is *planar* if no two edges intersect. A drawing of a digraph is *upward* if every edge is monotonically nondecreasing in the $y$-direction. A digraph is *upward planar* if it admits a planar upward drawing. Planarity and acyclicity are necessary but not sufficient conditions for upward planarity (see Fig. 1). We say that an embedded digraph is upward planar if it admits an upward planar drawing with the given embedding.

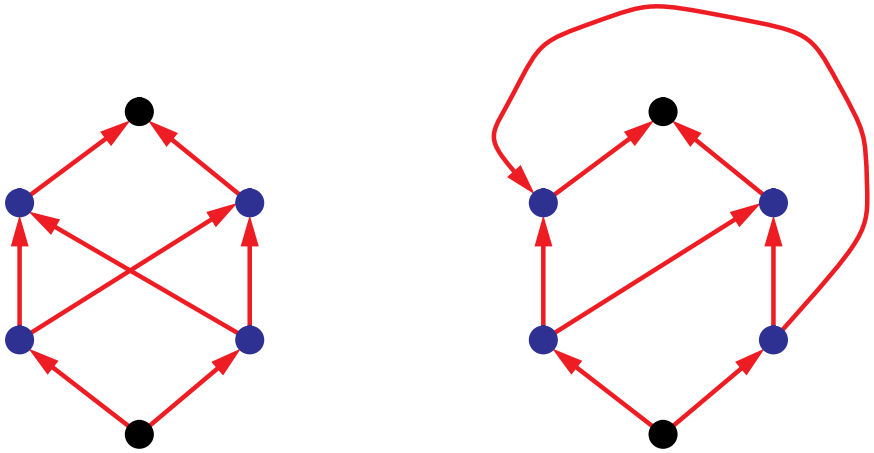


**Figure 1:** Two drawings of a planar acyclic digraph that is not upward planar: **(a)** an upward drawing which is not planar; **(b)** a planar drawing which is not upward.

We say that an ordered set is planar if its covering digraph (Hasse diagram) is upward planar. Testing upward planarity of digraphs and planarity of ordered sets are equivalent problems. Namely, it is easy to see that a digraph $\vec{G}$ is upward planar if and only if the ordered set $P$ with covering digraph $\vec{C}$ obtained from $\vec{G}$ by inserting a vertex along every transitive edge is planar.

Besides their applications to ordered sets, testing upward planarity and constructing upward planar drawings of digraphs are important for displaying hierarchical structures, such as subroutine-call graphs, is-a hierarchies, and PERT networks. While planarity testing can be done in linear time [19, 23, 8, 10], the complexity of upward planarity testing has been until recently an open problem.

# 3 Characterizing Upward Planarity

In this section we overview the known combinatorial characterizations of upward planarity, which make use of a variety of topological and geometric constructions. Some of these characterizations yield efficient upward planarity testing algorithms, as shown in Section 4.

## 3.1 Inclusion in a Planar *st*-Digraph

A simple characterization of upward planarity was independently discovered by Di Battista and Tamassia [13] and by Kelly [21]. We need the following definition: a *planar st-digraph*

is planar acyclic digraph with a single source, denoted $s$, a single sink, denoted $t$, and the edge $(s,t)$.

**Theorem 1 [13, 21]** *A digraph $\vec{G}$ is upward planar if and only it is a spanning subgraph of a planar st-digraph (i.e., edges can be added to $\vec{G}$ such that the resulting digraph $\vec{H}$ is a planar st-digraph).*

For example, the digraph of Fig. 2(a) is upward planar because adding to it some edges yields the planar $st$-digraph shown in Fig. 2(b).

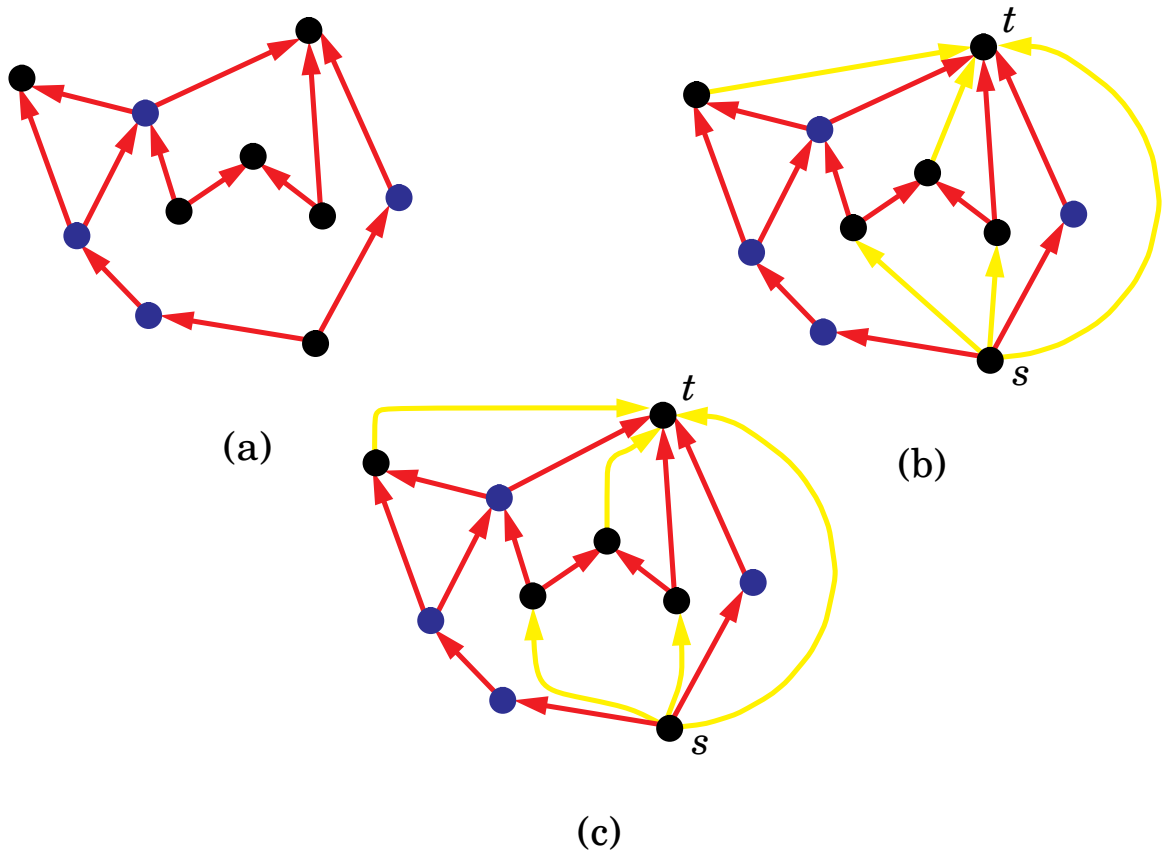


**Figure 2:** **(a)** An upward planar digraph $\vec{G}$. **(b)** Planar $st$-digraph obtained from $\vec{G}$ by adding the yellow edges. **(c)** Illustration of the proof of Theorem 1.

The if-part of Theorem 1 is immediate. Now, we give an intuitive illustration of the proof of the only-if part. Consider an upward planar drawing of $\vec{G}$, and call $s$ a source of $\vec{G}$ with lowest $y$-coordinate, and $t$ a sink of $\vec{G}$ with highest $y$-coordinate. At a sink $v \neq t$, we start drawing a new edge upwards. If we encounter an existing edge, we follow its path closely, as to avoid other edges, until we reach a vertex $w$ (see Fig. 2(c)). This creates a new edge $(v,w)$ that preserves planarity and acyclicity and "cancels" former sink $v$. This procedure needs a minor modification if a vertical ray emanating upwards from $v$ does not intersect any edge. By repeating this step, we are able to cancel all the sinks, except $t$. A similar procedure allows us to cancel all the sources, except $s$. Finally, we add the edge $(s,t)$, thus obtaining a planar $st$-digraph.

As shown in Section 4.1 the characterization of Theorem 1 provides an exponential-time upward planarity testing algorithm, thus placing the problem in NP.

Theorem 1 also implies that if a digraph admits an upward planar drawing, then it also admits an upward planar *straight-line* drawing. While the latter drawings are highly desirable for their simplicity, their area requirement may be prohibitive for visualization applications. Namely, Di Battista, Tamassia, and Tollis [15] show that there exists a family $\vec{G}_n$ of upward planar digraphs such that $\vec{G}_n$ has $2n+2$ vertices, and any planar straight-line upward drawing of $\vec{G}_n$ has area $\Omega(2^n)$. This result holds under any resolution rule that prevents drawings from being arbitrarily scaled down (e.g., integer vertex coordinates). The aforementioned worst-case exponential lower bound on the area holds also for the subclass of series-parallel digraphs [2].

## 3.2 Large Angles in Embedded Digraphs

This section presents the characterization of upward planarity for embedded digraphs given by Bertolazzi, Di Battista, Liotta, and Mannino [3, 4]. As shown in Section 4.2, this characterization yields an efficient upward planarity testing algorithm for embedded digraphs.

Let $\vec{G}$ be an embedded digraph. A vertex of $\vec{G}$ is *bimodal* if the cyclic sequence of its incident edges can be partitioned into two (possibly empty) linear sequences, one consisting of incoming edges and the other consisting of outgoing edges. The embedded digraph $\vec{G}$ is *bimodal* if all its vertices are bimodal. The following lemma can be easily proved by means of elementary geometric considerations:

**Lemma 1** *An embedded digraph is upward planar only if it is bimodal.*

Consider an assignment of labels from the set $\{small, large\}$ to the angles formed by pairs of incoming or outgoing edges of $\vec{G}$. The intuitive meaning of the labels is to indicate whether the angle is smaller or larger than $\pi$ in an upward planar straight-line drawing of $\vec{G}$. Let $p$ be either a vertex or a face of $\vec{G}$. We denote with $L(p)$ and $S(p)$ the number of angles of $p$ with label *large* and *small*, respectively. Simple geometric considerations show that the following two *consistency properties* must hold for any upward planar straight-line drawing of $\vec{G}$:

$$L(v) = \begin{cases} 0 & \text{if } v \text{ is an internal vertex,} \\ 1 & \text{if } v \text{ is a source or a sink.} \end{cases}$$

$$L(f) - S(f) = \begin{cases} -2 & \text{if } f \text{ is an internal face,} \\ +2 & \text{if } f \text{ is the external face.} \end{cases}$$

A labeling that verifies the consistency properties is illustrated in the example of Fig. 3.

Let $A(f)$ be number of angles in face $f$ formed by pairs of incoming edges ($A(f)$ is also equal to the number of angles in face $f$ formed by pairs of outgoing edges). The values $A(f)$ are determined by the embedding and are independent from the drawing (see Fig. 3). Clearly, for any labeling, we have:

$$L(f) + S(f) = 2\,A(f) \quad \text{for every face } f.$$

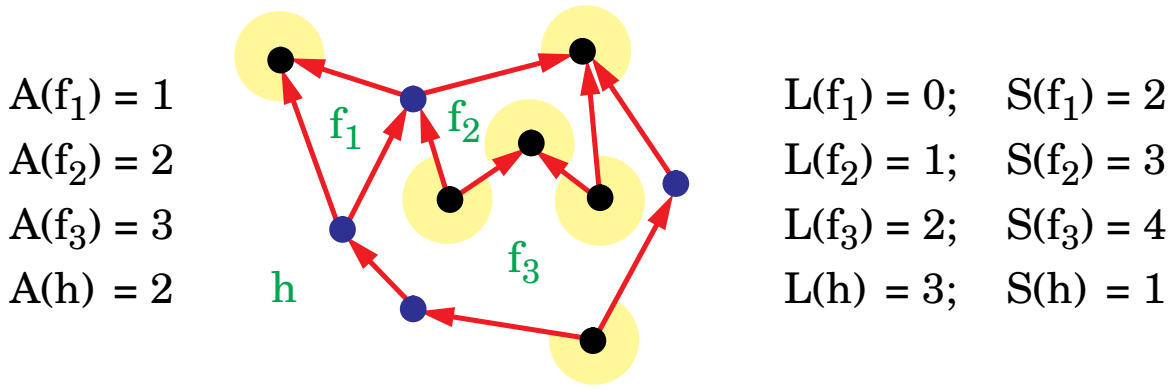


**Figure 3:** Large and small angles in an upward planar straight-line drawing. The large angles are shaded in yellow . Note that the consistency properties are verified by the labeling.

Thus, the consistency properties can be rewritten in the following format, where only large angles are taken into account

$$L(v) = \begin{cases} 0 & \text{if } v \text{ is an internal vertex,} \\ 1 & \text{if } v \text{ is a source or a sink.} \end{cases}$$

$$L(f) = \begin{cases} A(f) - 1 & \text{if } f \text{ is an internal face,} \\ A(f) + 1 & \text{if } f \text{ is the external face.} \end{cases}$$

Motivated by the above formulation, we consider now an assignment $\Phi$ that maps each vertex $v$ which is a source or a sink to a face $\Phi(v)$ incident on $v$, and we denote with $\Phi^{-1}(f)$ the set of vertices assigned to face $f$. We say that assignment $\Phi$ is *consistent* if there exists a face $h$ such that:

$$|\Phi^{-1}(f)| = \begin{cases} A(f) - 1 & \text{if } f \neq h, \\ A(f) + 1 & \text{if } f = h. \end{cases}$$

Assigning vertex $v$ to face $f = \Phi(v)$ corresponds to giving label *large* to the angle formed by $v$ in $f$, and a consistent assignment corresponds to a labeling with external face $h$ that verifies the consistency properties. In Fig. 4, we show the consistent assignment associated with the labeling of Fig. 3.

**Lemma 2 [3, 4]** *An embedded digraph is upward planar only if it admits a consistent assignment of sources and sinks to faces.*

As shown in [3, 4], the conditions given in Lemmas 1–2 are also sufficient.

**Theorem 2 [3, 4]** *An embedded digraph is upward planar if and only if it is acyclic, bimodal and admits a consistent assignment of sources and sinks to faces.*

Theorem 2 can be used to prove that the digraph of Fig. 1, which has a unique embedding, is not upward planar, as illustrated in Fig. 5.

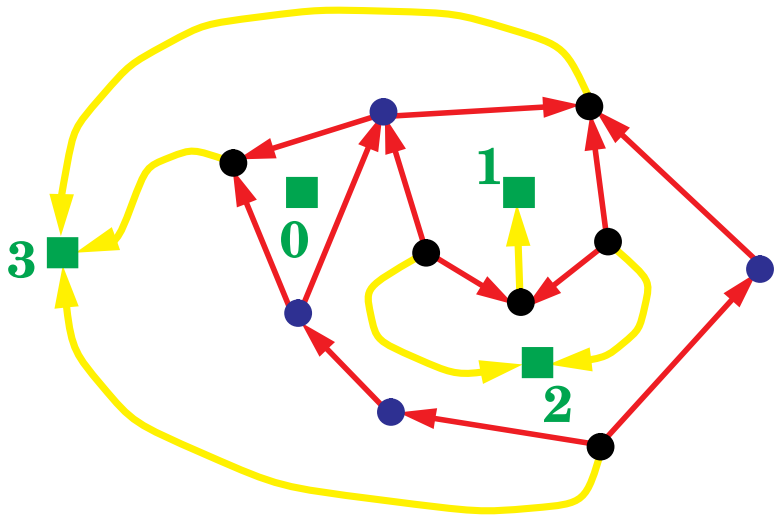


**Figure 4:** The consistent assignment, indicated with yellow arrows, associated with the labeling of Fig. 3. Small squares represent the faces. The quantity $A(f) - 1$ is shown next to each internal face $f$. The quantity $A(h) + 1$ is shown next to the external face $h$.

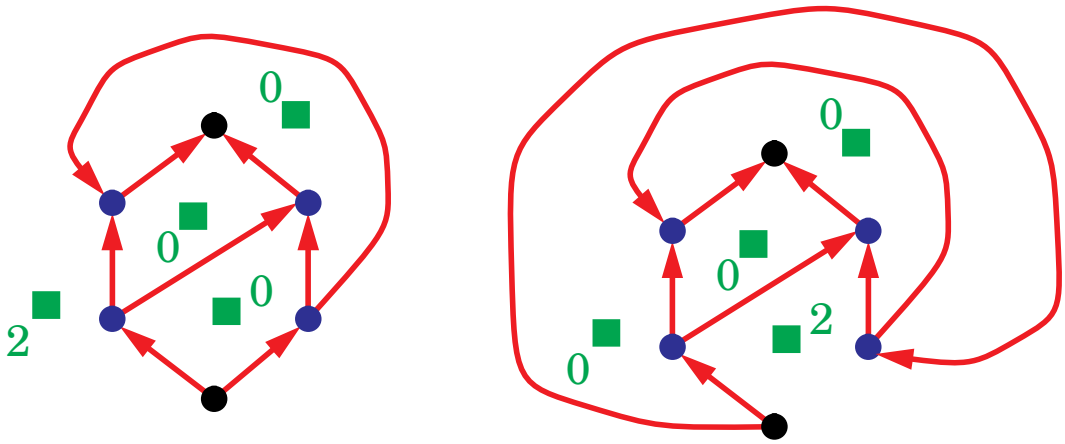


**Figure 5:** An embedded digraph that is bimodal but does not admit a consistent assignment of sources and sinks to faces, for any choice of the external face.

### 3.3 Forbidden Cycles for Single-Source Digraphs

Thomassen [27] characterizes the upward planarity of embedded digraphs with a single source in terms of certain forbidden (undirected) cycles. This characterization is used in the algorithm by Hutton and Lubiw [20].

**Theorem 3 [27]** *An embedded digraph $\vec{G}$ is upward planar if and only if it is acyclic and there is a choice of external face such that:*

- *the source of $\vec{G}$ is on the external face; and*
- *every undirected cycle $C$ of $\vec{G}$ has a vertex which becomes a sink after removing all the edges outside $C$.*

Note that the choice of external face determines what is inside and outside a cycle. The necessity of the condition is easy to verify by considering the vertex of $C$ with highest $y$-coordinate in an upward planar drawing of $\vec{G}$ (see, e.g., Fig. 3).

Theorem 3 provides an alternative proof of the fact that the digraph of Fig. 1, which has a unique embedding, is not upward planar, as shown in Fig. 6.

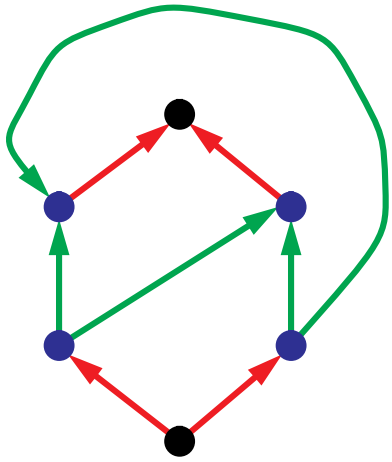

**Figure 6:** Example of forbidden cycle (shown with green lines) in the embedded digraph of Fig. 1(b).

## 3.4 Forests in Single-Source Embedded Digraphs

Given an embedded digraph $\vec{G}$ with a single source $s$, we define the undirected *face-sink graph* $\mathcal{F}$ of $\vec{G}$ as follows:

- The vertices of $\mathcal{F}$ are the faces of $\vec{G}$ and the vertices of $\vec{G}$ that are sinks for at least one face of $\vec{G}$.
- $\mathcal{F}$ has an edge $(f, v)$ if and only if vertex $v$ is a sink of face $f$.

The edges of graph $\mathcal{F}$ incident on a vertex $v$ which is a sink of $\vec{G}$ represent the possible assignments of $v$ to the faces of $\vec{G}$ (see Section 3.2). In Fig. 7(b), we show the face-sink graph $\mathcal{F}$ of the embedded single-source digraph $\vec{G}$ of Fig. 7(a). Note that, in this example where $\vec{G}$ is upward planar, $\mathcal{F}$ is a forest.

Fig. 7(c) shows a consistent assignment of the source and sinks of $\vec{G}$ to the faces. This assignment corresponds to rooting each tree of $\mathcal{F}$ and orienting its edges from child to parent. Namely, a sink $v$ is assigned to a face $f$ if and only if $v$ is a child of $f$ in the rooted tree. containing $v$ and $f$. We also notice that the roots of all but one tree (denoted with $\mathcal{T}$) are internal vertices of $\vec{G}$. Tree $\mathcal{T}$ does not contain any internal vertex of $\vec{G}$ and is rooted at the external face $h$.

The example of Fig. 7 is an illustration of the following theorem.

**Theorem 4** [5] *Let $\vec{G}$ be an embedded single-source digraph, and $h$ a face of $\vec{G}$. Digraph $\vec{G}$ is upward planar, subject to $h$ being the external face, if and only if all the following conditions are satisfied:*

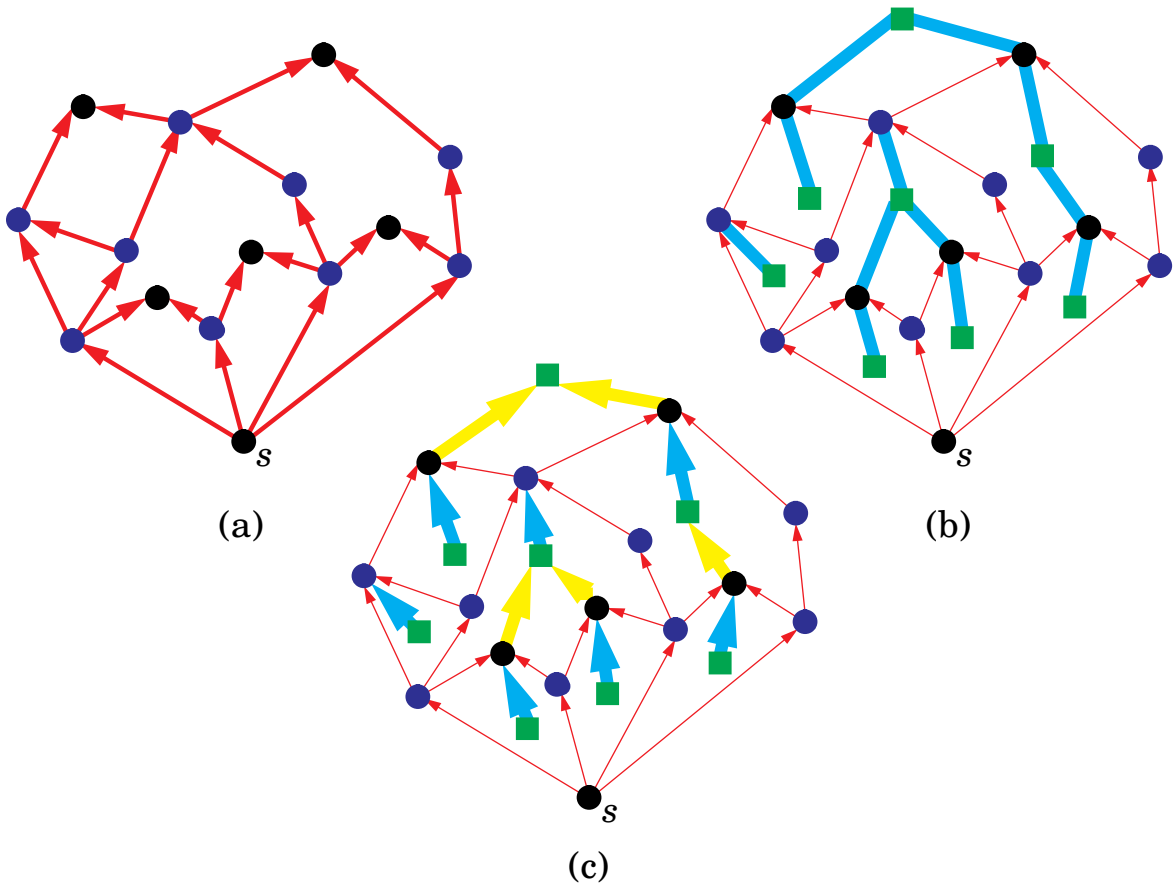


**Figure 7:** **(a)** An embedded single-source digraph $\vec{G}$. **(b)** The face-sink graph $\mathcal{F}$ of $\vec{G}$, shown with light-blue edges. **(c)** A consistent assignment of the source and sinks of $\vec{G}$ to its faces (shown with yellow edges), and the corresponding rooting of the trees of $\mathcal{F}$.

1. *the source of $\vec{G}$ is on the boundary of face $h$.*
2. *the face-sink graph $\mathcal{F}$ of $\vec{G}$ is a forest;*
3. *a tree $\mathcal{T}$ of $\mathcal{F}$ has no internal vertices of $\vec{G}$, while the remaining trees have exactly one internal vertex; and*
4. *face $h$ is a vertex of tree $\mathcal{T}$.*

An additional example for Theorem 4 is given in Fig. 8. While Conditions 1–3 of Theorem 4 are satisfied, Condition 4 is not satisfied because face $h$ is in a tree of $\mathcal{F}$ that contains an internal vertex.

## 3.5 Forbidden Structures for Lattices

An important consequence of the fact that the digraph $\vec{G}$ of Fig. 1 is not upward planar is that an upward planar digraph cannot contain a subgraph homeomorphic to $\vec{G}$. This observation implies the following result.

**Theorem 5 [6, p. 32, Ex. 7(a)]** *The ordered set induced by an upward planar digraph with one source and one sink is a lattice, i.e., a bounded planar ordered set is a lattice.*

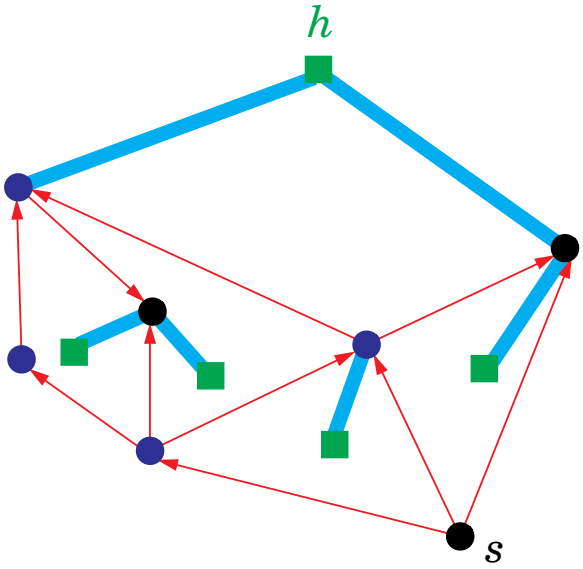


**Figure 8:** An embedded single-source digraph that is not upward planar and its face-sink graph.

Kelly and Rival [22] give a Kuratowski-type characterization of planar lattices in terms of forbidden sublattices. Namely, they define a family $\mathcal{L}$ of nonplanar lattices (see Fig. 9) and show that $\mathcal{L}$ is the minimal obstruction set, under sublattice containment, for upward planarity.

**Theorem 6** [22] *A lattice is planar if and only if it contains no (order) sublattice isomorphic to a lattice from the family $\mathcal{L}$. Moreover, $\mathcal{L}$ is the minimum such family.*

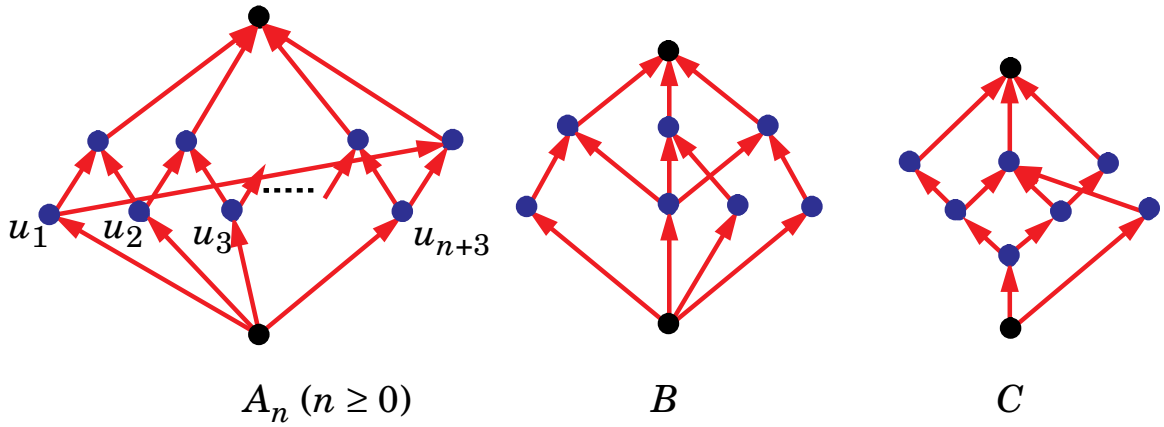


**Figure 9:** Some of the nonplanar lattices in the set $\mathcal{L}$ of Theorem 6.

Platt [25] characterizes the planarity of a lattice in terms of the planarity of an undirected graph related to its covering digraph.

**Theorem 7** [25] *A lattice is planar if and only if the undirected graph obtained from its covering digraph by ignoring the orientation of the edges and adding an edge between the source and sink is planar.*

Lempel, Even, and Cederbaum [23] relate the planarity of biconnected undirected graphs to the upward planarity of acyclic digraphs with one source and one sink. An *st-orientation* of an undirected graph $G$ is an acyclic digraph with exactly one source $s$ and one sink $t$, joined by an edge, which is obtained from $G$ by orienting its edges. A biconnected graph admits an $st$-orientation for any edge $(s,t)$.

**Theorem 8** **[23]** *Let $G$ be a biconnected graph, and $\vec{G}$ be an st-orientation of $G$. We have that $G$ is planar if and only if $\vec{G}$ is upward planar.*

The planarity of lattices is also characterized by the (order) dimension.

**Theorem 9** **[1]** *A lattice is planar if and only if it has dimension at most two.*

### 3.6 Some Classes of Upward Planar Digraphs

Several classes of digraphs have been identified whose members are always upward planar.

**Theorem 10** *All the digraphs in the following classes are upward planar:*

- *digraphs whose underlying undirected graph is a forest;*
- *planar st-digraphs;*
- *series-parallel digraphs;*
- *planar bipartite digraphs* [12].

## 4 Testing Upward Planarity

In this section, we overview the known upward planarity testing algorithms. Since the problem is in general NP-complete (see Section 5), efficient algorithms exist only for special classes of digraphs.

### 4.1 General Digraphs

Testing whether an $n$-vertex digraph is a planar $st$-digraph can be easily done in $O(n)$ time by testing separately its acyclicity, planarity (as an undirected graph), and the existence of a single source and single sink, joined by an edge. Acyclicity can be tested in $O(n)$ time by means of an elementary depth-first-search method (see, e.g., [9]). Planarity can also be tested in $O(n)$ time (see, e.g., [19]).

Hence, Theorem 1 yields an exponential-time (albeit linear-space) upward planarity testing algorithm, which consists of adding all the possible subsets of at most $3n-6$ edges, and testing whether each of the resulting digraphs is a planar $st$-digraph.

**Theorem 11** *Upward planarity testing is in NP.*

See [26] for an alternative proof of Theorem 11.

## 4.2 Embedded Digraphs

In this section we present the algorithm by Bertolazzi, Di Battista, Liotta and Mannino [3, 4] for testing whether an embedded digraph is upward planar. The algorithm is based on the characterization of Theorem 2.

To test whether an embedded digraph $\vec{G}$ admits a consistent assignment of sources and sinks to faces for a given choice of external face $h$, we construct a bipartite flow network $\mathcal{B}(h)$ as follows (see Fig. 10):

- The vertices of network $\mathcal{B}(h)$ are the sources, sinks and faces of $\vec{G}$. The sources and sinks of $\vec{G}$ are the sources of network $\mathcal{B}(h)$ and supply each a unit of flow. Each face $f$ of $\vec{G}$ is a sink of network $\mathcal{B}(h)$, with demand equal to $A(f) - 1$ units of flow if $f \neq h$, and equal to $A(f) + 1$ units of flow if $f = h$.
- Network $\mathcal{B}(h)$ has an edge $(v, f)$ if $v$ is a source or sink of $\vec{G}$ on face $f$.

A *flow* in network $\mathcal{B}(h)$ is an assignment of values 0 or 1 to the edges of $\mathcal{B}(h)$ such that, for each source $v$ of $\mathcal{B}(h)$, the sum of the values assigned to the outgoing edges of $v$ is less than or equal to the supply of $v$, and for each sink $f$ of $\mathcal{B}(h)$, the sum of the values assigned to the incoming edges of $f$ is less than or equal to the demand of $f$. The value of a flow in $\mathcal{B}(h)$ is the sum of the values assigned to its edges.

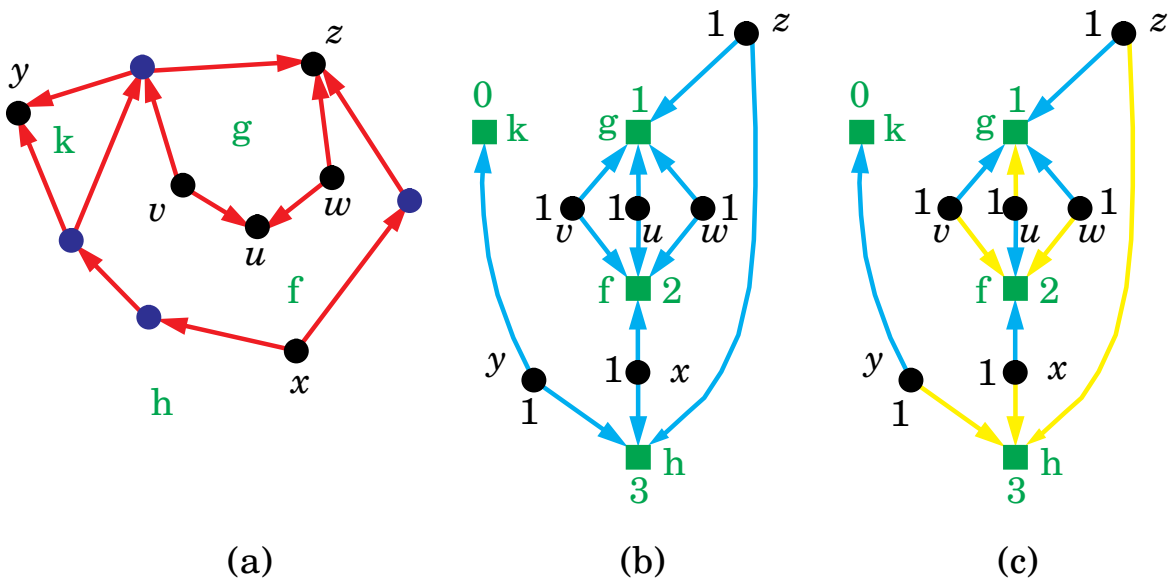


**Figure 10:** **(a)** The embedded digraph $\vec{G}$ of Fig. 2. **(b)** The network $\mathcal{B}(h)$ associated with $\vec{G}$ and external face $h$. The vertices of $\mathcal{B}(h)$ are labeled with their supplies or demands. **(c)** Example of flow in network $\mathcal{B}(h)$. The yellow edges have unit flow and give a consistent assignment of sources and sinks to faces.

**Lemma 3 [3, 4]** *The bipartite flow network $\mathcal{B}(h)$ associated with the embedded digraph $\vec{G}$ and face $h$ has $O(r)$ vertices, where $r$ is the number of sources and sinks of $\vec{G}$. Also, $\vec{G}$ admits a consistent assignment of sources and sinks to faces, subject to $h$ being the external face, if and only if $\mathcal{B}(h)$ admits a flow of value $r$.*

Constructing network $\mathcal{B}(h)$ takes $O(n)$ time, where $n$ is the number of vertices of $\vec{G}$. The existence of a flow for $\mathcal{B}(h)$ can be tested in $O(r^2)$ time by means of $r$ flow augmentations (for the standard augmenting path method to solve network flow problems, see, e.g., [9]). Hence, we can test the upward planarity of $\vec{G}$ in $O(nr^2)$ time by repeating the above procedure for all the $O(n)$ possible choices of the external face $h$.

As shown in [4], the time complexity can be reduced to $O(n + r^2)$ using the following algorithm.

1. Construct flow network $\mathcal{B}$, which is the same as $\mathcal{B}(h)$, except that each face $f$ has demand $A(f) - 1$. Note that flow network $\mathcal{B}$ is independent from the choice of external face.
2. Test whether $\mathcal{B}$ admits a flow of value $r - 2$. If not, then return "not upward planar".
3. For each face $f$ which is a sink of $\mathcal{B}$, increase its demand by two units, and test whether the flow of value $r - 2$ in $\mathcal{B}$ can be augmented by two units.
4. If the augmentation test of Step 3 is successful for at least one face, return "upward planar" and report all the faces for which it is successful. Otherwise, return "not upward planar".

Let $n$ be the number of vertices of $\vec{G}$, and let $r$ be the number of its sources and sinks. Step 1 takes $O(n)$ time. Step 2 can be performed with $r - 2$ flow augmentations, each taking $O(r)$ time, and hence runs in $O(r^2)$ time. In Step 3, at most two augmentations are performed for each face. Hence, Step 1 takes $O(r^2)$ time. Finally, Step 4 takes $O(r)$ time.

By Theorem 2, the complete upward planarity test algorithm for an embedded digraph $\vec{G}$ consists of testing first whether $\vec{G}$ is acyclic and bimodal, which takes $O(n)$ time, and then testing whether $\vec{G}$ admits a consistent assignment of sources and sinks to its faces, which takes $O(n + r^2)$ time.

**Theorem 12** [4] *Let $\vec{G}$ be an embedded digraph with $n$ vertices and $r$ sources and sinks. We can test whether $\vec{G}$ is upward planar in $O(n + r^2) = O(n^2)$ time.*

Since a triconnected planar digraph has a unique embedding, we obtain the following corollary.

**Theorem 13** [4] *Let $\vec{G}$ be a triconnected planar digraph with $n$ vertices and $r$ sources and sinks. We can test whether $\vec{G}$ is upward planar in $O(n + r^2) = O(n^2)$ time.*

## 4.3 Embedded Single-Source Digraphs

Hutton and Lubiw [20] show how to apply Theorem 3 to test the upward planarity of an embedded single-source digraph with $n$ vertices in $O(n^2)$ time.

Theorem 4 yields the following algorithm for testing the upward planarity of an embedded single-source digraph $\vec{G}$ [5].

1. Test whether $\vec{G}$ is acyclic and bimodal. If either of these properties is not verified, return "not upward planar".
2. Construct the face-sink graph $\mathcal{F}$ of $\vec{G}$ (see Section 3.4).
3. Check Conditions 2 and 3 of Theorem 4. If these conditions are not verified, then return "not upward planar", else let $\mathcal{T}$ be the tree of Condition 3.

4. Report the set of faces of $\vec{G}$ that contain the source $s$ of $\vec{G}$ (Condition 1) and are vertices of tree $\mathcal{T}$ (Condition 4). If this set is empty, then return "not upward planar" else return "upward planar".

The above algorithm can be easily implemented in $O(n)$ time.

**Theorem 14** [5] *Let $\vec{G}$ be an embedded single-source digraph with $n$ vertices. We can test whether $\vec{G}$ is upward planar in $O(n)$ time.*

By applying Theorem 14 to the embedded digraph $\vec{G}$ of Fig. 8, we conclude that $\vec{G}$ is not upward planar (for any choice of the external face).

## 4.4 Single-Source Digraphs

In this section we present a decomposition strategy for testing the upward planarity of single-source digraphs, which was originally proposed by Hutton and Lubiw [20], and later improved by Bertolazzi, Di Battista, Mannino, and Tamassia [5].

To start with, it is easy to prove that the biconnected components of a single-source digraph can be separately tested.

**Lemma 4** [20] *A connected single-source digraph $\vec{G}$ is upward planar if and only all the biconnected components of $\vec{G}$ are upward planar.*

By Lemma 4, we restrict our attention to a biconnected single-source digraph $\vec{G}$. The basic idea is to decompose $\vec{G}$ into its triconnected components [18] and test each of them separately for upward planarity, with certain constraints imposed on them to ensure that merging back together the triconnected components preserves upward planarity. Recall that the upward planarity of a triconnected digraph can be efficiently tested (Theorems 13 and 14).

The decomposition is performed by splitting the digraph $\vec{G}$ into split-components with respect to a separation pair, and recursively decomposing each split-component. Each split component is constrained by attaching to it a certain digraph, called *marker*, which represents the rest of the digraph. There are four type of markers, as shown in Fig. 11.

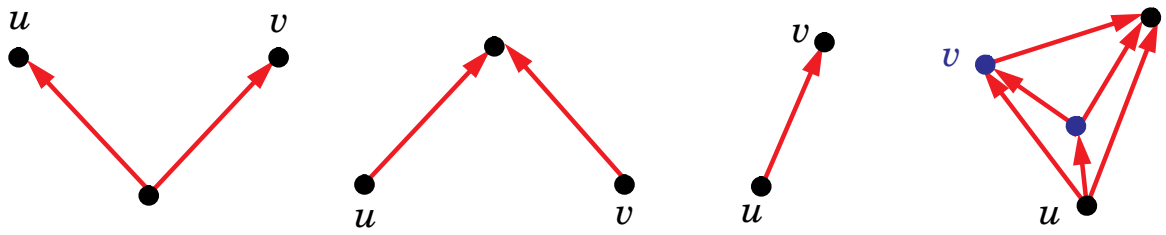


**Figure 11:** Markers.

Suppose that $\vec{G}$ has two split-components, $\vec{H}$ and $\vec{K}$, with respect to a separation pair $\{u, v\}$. We augment each of $\vec{H}$ and $\vec{K}$ with an appropriate marker that represents the other split-component, and recursively test each of them for upward planarity. The markers have a twofold purpose:

- when testing a split-component for upward planarity, they take into account the "shape" of the other split-component; and

- they ensure that each digraph in the decomposition process is biconnected and has a single source.

For example, consider the biconnected single-source digraph $\vec{G}$ shown in Fig. 12(a), which is not upward planar. The split-components of $\vec{G}$ with respect to $\{u, v\}$, shown in Fig. 12(b,c), are each upward planar. By adding the appropriate markers, we obtain the augmented split-components shown in Fig. 12(d,e). Note that the augmented split-component of Fig. 12(e) is not upward planar, as it can be shown by applying Theorem 14. This indicates that $\vec{G}$ is not upward planar.

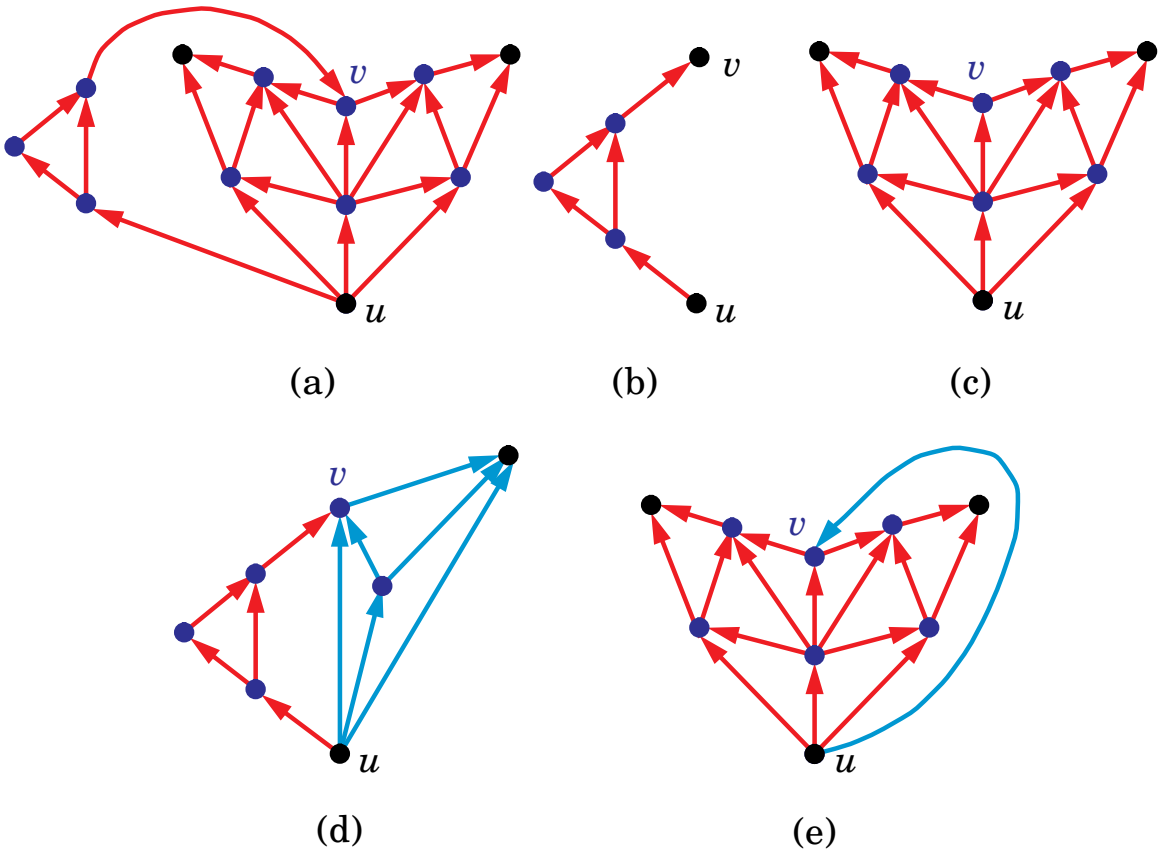


**Figure 12:** **(a)** A biconnected single-source digraph $\vec{G}$ with a separation pair $u, v$. **(b)** Split-component $\vec{H}$ of $\vec{G}$ with respect to $u, v$. **(c)** Split-component $\vec{K}$ of $\vec{G}$ with respect to $u, v$. **(d)** Split-component $\vec{H}$ augmented with the appropriate marker (light blue). **(e)** Split-component $\vec{K}$ augmented with the appropriate marker (light blue).

In addition to the markers, we must take into account the inside-outside relationship of the split components with respect to the external face. For example, if we determine that in any upward planar drawing of a split-component the separation-pair does not appear on the external face, then we must test the other split-component for upward planarity, subject to the separation-pair appearing on the external face. Hence, we should keep track in the decomposition process of vertices that are constrained to appear on the external face.

Hutton and Lubiw [20] present an $O(n^2)$-time upward planarity testing algorithm for single-source digraphs, based on the above decomposition strategy. Bertolazzi, Di Battista,

Mannino, and Tamassia [5] reduce the time complexity to $O(n)$ by combining a linear-time algorithm for triconnected single source-digraphs (Theorem 14) with an improved decomposition strategy, which is outlined below.

The SPQR-tree [14] $T(G)$ of a biconnected graph (or digraph) $G$ describes the arrangement of its triconnected components (see Fig. 13). $T(G)$ is an unrooted tree whose nodes are of four types: S, P, Q, and R. Each edge of $T(G)$ is associated with an edge or separation pair of $G$, and each node $\mu$ of $T(G)$ is associated with a multigraph, called the *skeleton* of $\mu$. A P-node corresponds to a parallel composition of split-components with respect to a separation pair (see Fig. 14(a)), and its skeleton is a bundle of parallel edges. An S-node corresponds to a cyclic arrangement of split-components and separation pairs (see Fig. 14(b)), and its skeleton is a cycle. An R-node corresponds to a maximal triconnected homeomorphic subgraph of $G$, which is its skeleton (see Fig. 13(c)). A Q-node corresponds to an edge of $G$, and its skeleton consists of two parallel edges. Note that the edge directions are ignored in the definition of SPQR-tree.

The upward planarity testing algorithm of [5] first replaces each edge of the skeletons by a gadget (playing the same role as the marker of [20]), which is either a directed edge or a *peak* (see Fig. 15), and tests each augmented skeleton for upward planarity, marking those gadgets that can appear on the external face in a planar upward drawing.

The existence of a compatible inside-outside relationship between skeletons is then tested using the SPQR-tree. Namely, the nesting of the skeletons is represented by a rooting of the SPQR-tree $T(\vec{G})$: if node $\nu$ is the parent of node $\mu$, then the skeleton of $\nu$ is "outer" and the skeleton of $\mu$ "inner" in the embedding of $\vec{G}$. Of course, this can be done only if the gadget of the skeleton of $\mu$ representing the split-component of the skeleton of $\nu$ is marked. Also, the source $s$ of $\vec{G}$ must appear in the outermost skeleton (i.e., at the root of $T(\vec{G})$).

In the example of Fig 13, only the skeleton of node $\mu_3$ contains the source $s$. Hence, the SPQR-tree must be rooted at $\mu_3$. However, this implies that the gadget $e$ of the skeleton of node $\mu_2$ associated with the split-component of the skeleton of $\mu_3$ must appear on the external face in an upward planar drawing. This is not possible, as it can be shown by applying Theorem 14. We conclude that the single-source digraph of Fig 13(a) is not upward planar.

**Theorem 15 [5]** *Let $\vec{G}$ be a single-source digraph with $n$ vertices. We can test whether $\vec{G}$ is upward planar in $O(n)$ time.*

## 4.5 Outerplanar Digraphs

Papakostas [24] has recently shown that upward planarity testing can be efficiently performed for outerplanar digraphs. His algorithms exploit the fact that the dual of an embedded outerplanar graph is a tree.

**Theorem 16 [24]** *Let $\vec{G}$ be an embedded outerplanar digraph with $n$ vertices. We can test whether $\vec{G}$ is upward planar in $O(n)$ time.*

**Theorem 17 [24]** *Let $\vec{G}$ be an outerplanar digraph with $n$ vertices. We can test whether $\vec{G}$ is upward planar in $O(n^2)$ time.*

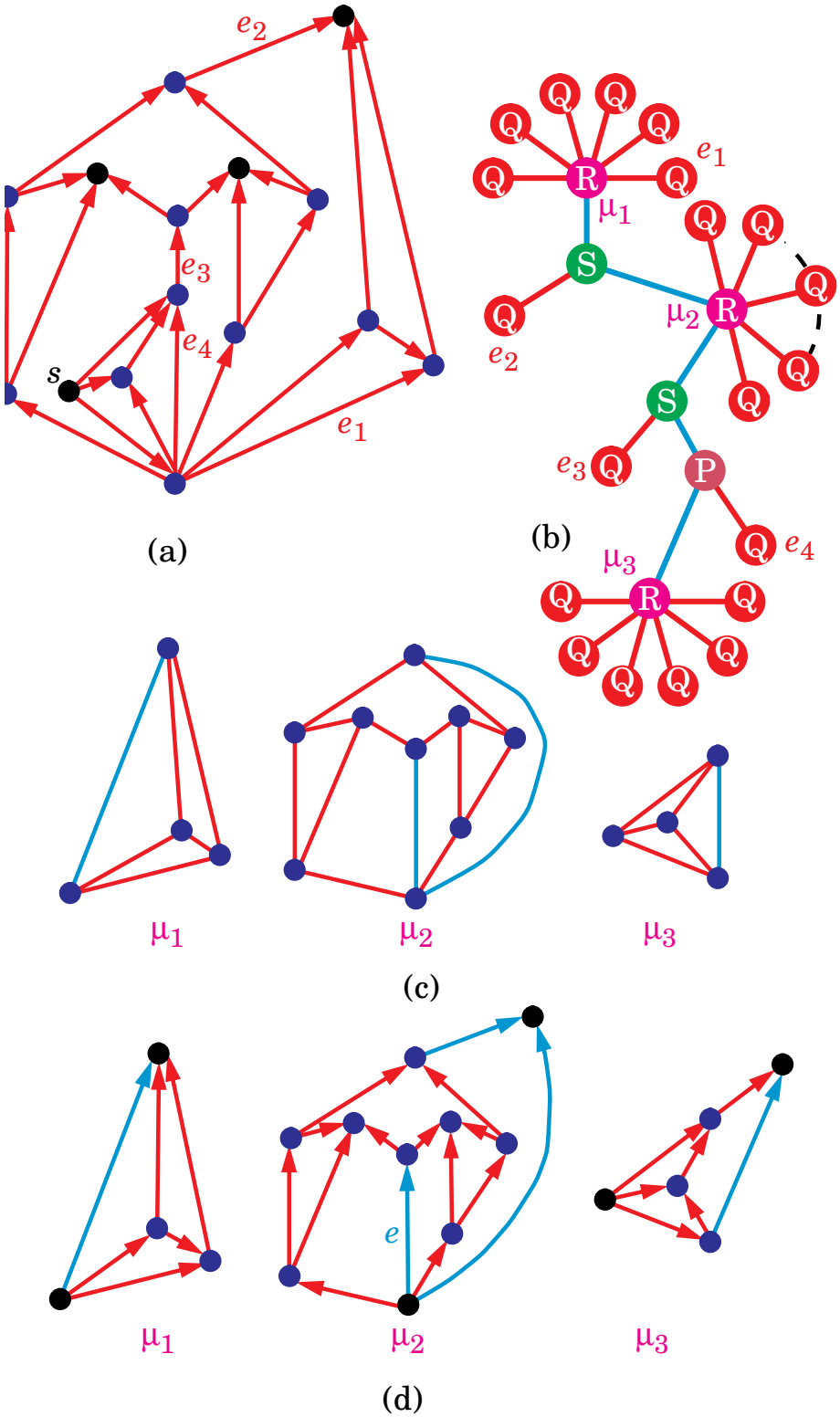


**Figure 13:** **(a)** A biconnected single-source digraph $\vec{G}$. **(b)** The SPQR-tree $T(\vec{G})$ of $\vec{G}$. The edges of $T(\vec{G})$ that correspond to the edges and separation pairs of $\vec{G}$ are shown as red and light blue lines respectively. **(c)** Skeletons of the R-nodes of $T(\vec{G})$. The light blue edges correspond to split-components. **(d)** Augmented skeletons of the R-nodes of $T(\vec{G})$. The gadgets representing split-components are shown by light blue lines.

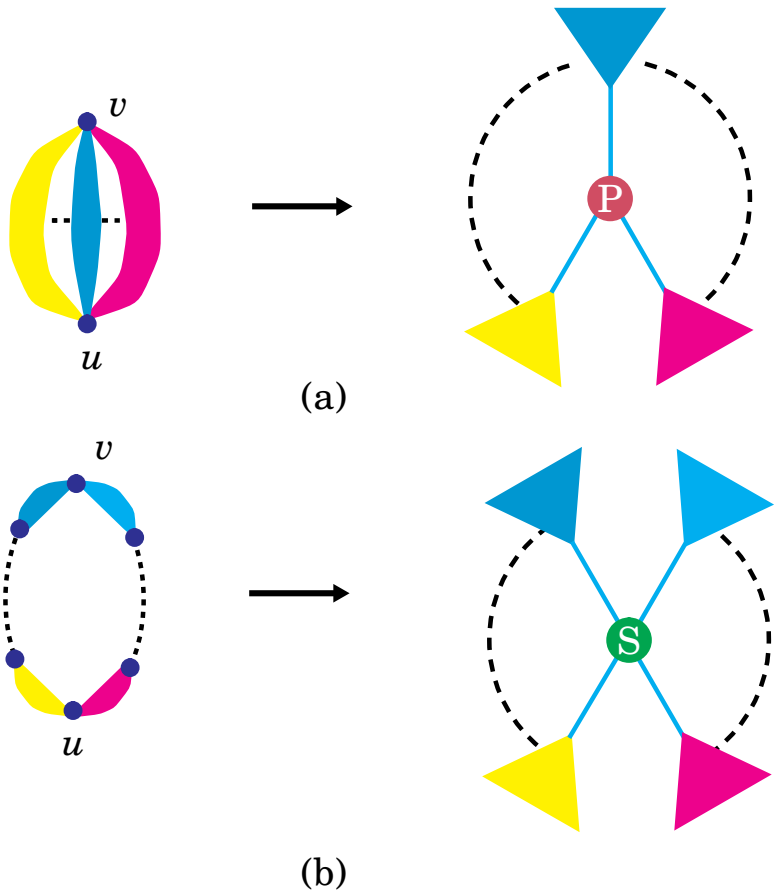


**Figure 14:** **(a)** Schematic illustration of a P-node. **(b)** Schematic illustration of an S-node.

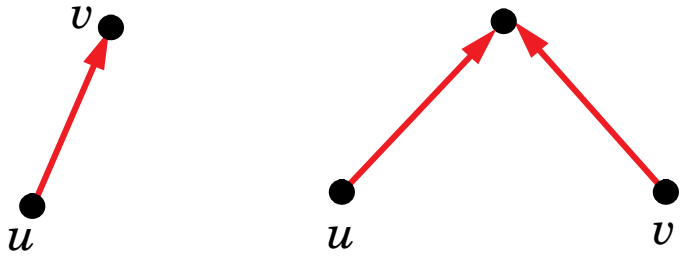


**Figure 15:** The edge and peak gadgets.

# 5 Upward Planarity Testing is NP-complete

In this section we overview the proof by Garg and Tamassia [17] that upward planarity testing is NP-complete. We assume standard concepts and definitions on NP-completeness [16]. The proof is a reduction from the following well-known NP-complete problem:

NOT-ALL-EQUAL-3-SAT Given a set of clauses with three literals each, is there a truth assignment such that each clause has at least one true literal and one false literal?

Section 5.1 describes some special graphs called *tendrils* and *wiggles* that are used in the reduction. A reduction from NOT-ALL-EQUAL-3-SAT to an auxiliary flow problem is given

in Section 5.2. Section 5.3 describes the reduction from the auxiliary flow problem to the upward planarity testing problem.

## 5.1 Tendrils and Wiggles

In this section, we define several digraphs that will be used as gadgets in our reductions.

We show in Fig. 16(a) *tendril* $T_k$ ($k \geq 1$), which is an acyclic digraph with $k+1$ sources and $k+1$ sinks. We also define tendril $T_0$ as a a digraph consisting of a single edge. Tendril $T_k$ ($k \geq 0$) has a designated source and a designated sink, called the *poles* of $T_k$. We shall consider transformations where a directed edge $(u, v)$ of a digraph is replaced with a tendril $T_k$, where the source is identified with $u$ and the sink with $t$.

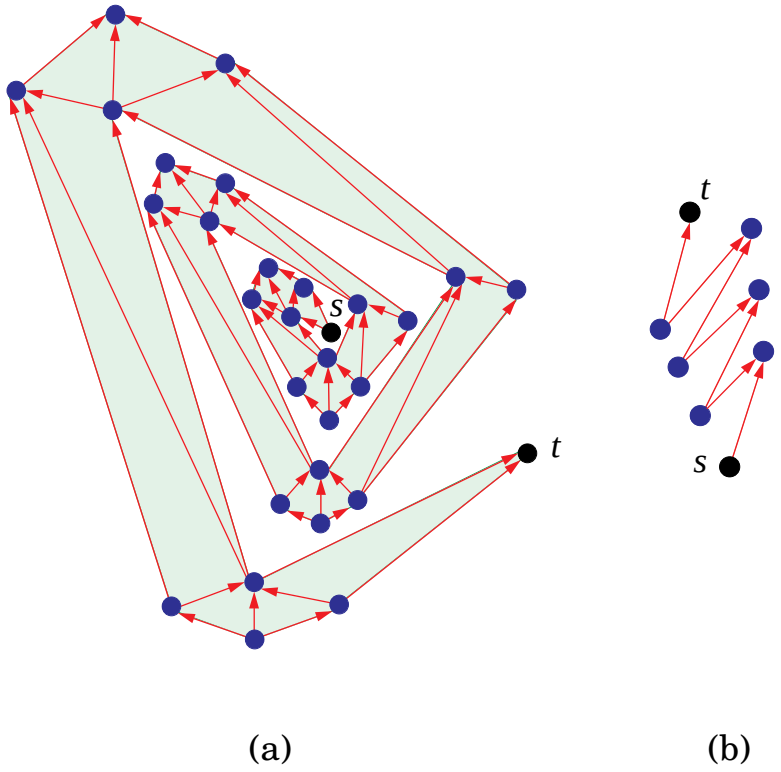


**Figure 16:** **(a)** Tendril $T_3$. **(b)** Wiggle $W_3$.

**Lemma 5** [17] *Tendril $T_k$ is upward planar and admits a unique upward planar embedding.*

In the upward planar embedding of $T_k$, the external face consists of two paths between $s$ and $t$. One such path, called *outer path*, has $2k$ large angles and no small angles, and the other path, called *inner path*, has $2k$ small angles and no large angles. When a tendril replaces an edge of an embedded planar digraph, the outer path becomes a subpath of a face, and we say that the *contribution* of the outer path to the face is $+2k$. Similarly, we say that the contribution of the inner path to its face is $-2k$.

Figure 16(b) shows a *wiggle* $W_k$, which is an acyclic digraph consisting of a chain of $2k+1$ edges whose orientation alternates along the chain. The extreme vertices of $W_k$, a source and a sink, are called the *poles* of $W_k$. We shall consider transformations where a directed

edge $(u, v)$ of an embedded digraph is replaced with wiggle $W_k$, where $s$ is identified with $u$ and $v$ with $t$. Given an upward embedding of $W_k$, we say that the *contribution* of $W_k$ to a face $f$ containing $W_k$ is the number of large angles minus the number of small angles of $W_k$ in $f$. Clearly, $W_k$ can be upward embedded to give to $f$ any contribution $2i$ with $0 \leq i \leq k$. Note that if $G_k$ gives contribution $c$ to a face, it gives contribution $-c$ to the other face it belongs to.

## 5.2 An Auxiliary Undirected Flow Problem

In this section we define two auxiliary flow problems and show that they are equivalent to NOT-ALL-EQUAL-3-SAT under polynomial-time reductions.

A *switch-flow network* is an undirected flow network $\mathcal{N}$ where each edge is labeled with a range $[c' \cdots c'']$ of nonnegative integer values, called the *capacity range* of the edge. For simplicity, we denote the capacity range $[c \cdots c]$ with $[c]$. A *flow* for a switch-flow network is an orientation of and an assignment of integer "flow" values to the edges of the network. A *feasible flow* is a flow that satisfies the following two properties:

*Range property:* the flow assigned to an edge is an integer within the capacity range of the edge.

*Conservation property:* the total flow entering a vertex from the incoming edges is equal to the total flow exiting the vertex from the outgoing edges.

Starting from an instance $\mathcal{S}$ of NOT-ALL-EQUAL-3-SAT, we construct a switch-flow network $\mathcal{N}$ as follows (see Fig. 17). Let the literals of $\mathcal{S}$ be denoted with $x_1, y_1, \cdots, x_n, y_n$, where $y_i = \overline{x_i}$ , and the clauses of $\mathcal{S}$ be denoted with $c_1, \cdots, c_m$. Let $\theta$ be a positive integer parameter. We denote with $\alpha_i$ and $\beta_i$ $(i = 1, \cdots, n)$ the number of occurrences of literals $x_i$ and $y_i$ in the clauses of $\mathcal{S}$, respectively. Note that $\sum_{i=1}^{n}(\alpha_i + \beta_i) = 3m$. Also, we define $\gamma_i = (2i - 1)\theta$ and $\delta_i = 2i\theta$ $(i = 1, \cdots, n)$. Network $\mathcal{N}$ has a *literal vertex* for each literal of $\mathcal{S}$ and a *clause vertex* for each clause of $\mathcal{S}$, plus a special dummy vertex $z$. There are three types of edges in $\mathcal{N}$:

*Literal edges* joining pairs of literals associated with the same boolean variable. The capacity range of literal edge $(x_i, y_i)$ is $[\alpha_i\gamma_i + \beta_i\delta_i]$.

*Clause edges* joining each literal to each clause. The capacity range of clause edge $(x_i, c_j)$ is $[\gamma_i]$ if $x_i \in c_j$, and $[0]$ otherwise. The capacity range of clause edge $(y_i, c_j)$ is $[\delta_i]$ if $y_i \in c_j$, and $[0]$ otherwise.

*Dummy edges* joining each literal and each clause to the dummy vertex. The capacity ranges of dummy edges $(z, x_i)$ and $(z, y_i)$ are $[\beta_i\delta_i]$ and $[\alpha_i\gamma_i]$, respectively. The capacity range of dummy edge $(z, c_j)$ is $[0 \cdots \eta_j - 2\theta]$, where $\eta_j$ is the sum of the capacities of the clause edges incident on $c_j$.

The construction of network $\mathcal{N}$ from $\mathcal{S}$ is straightforward and can be carried out in $O(nm)$ time.

A feasible flow in network $\mathcal{N}$ corresponds to a satisfying truth assignment for $\mathcal{S}$. Namely, we have that a literal is true whenever its incident literal edge is incoming in the feasible flow (see Fig. 17(b)) and its incident clause edges with nonzero capacity range are outgoing. Also, the three clause edges with nonzero capacity range incident on a clause vertex $c_j$ cannot be

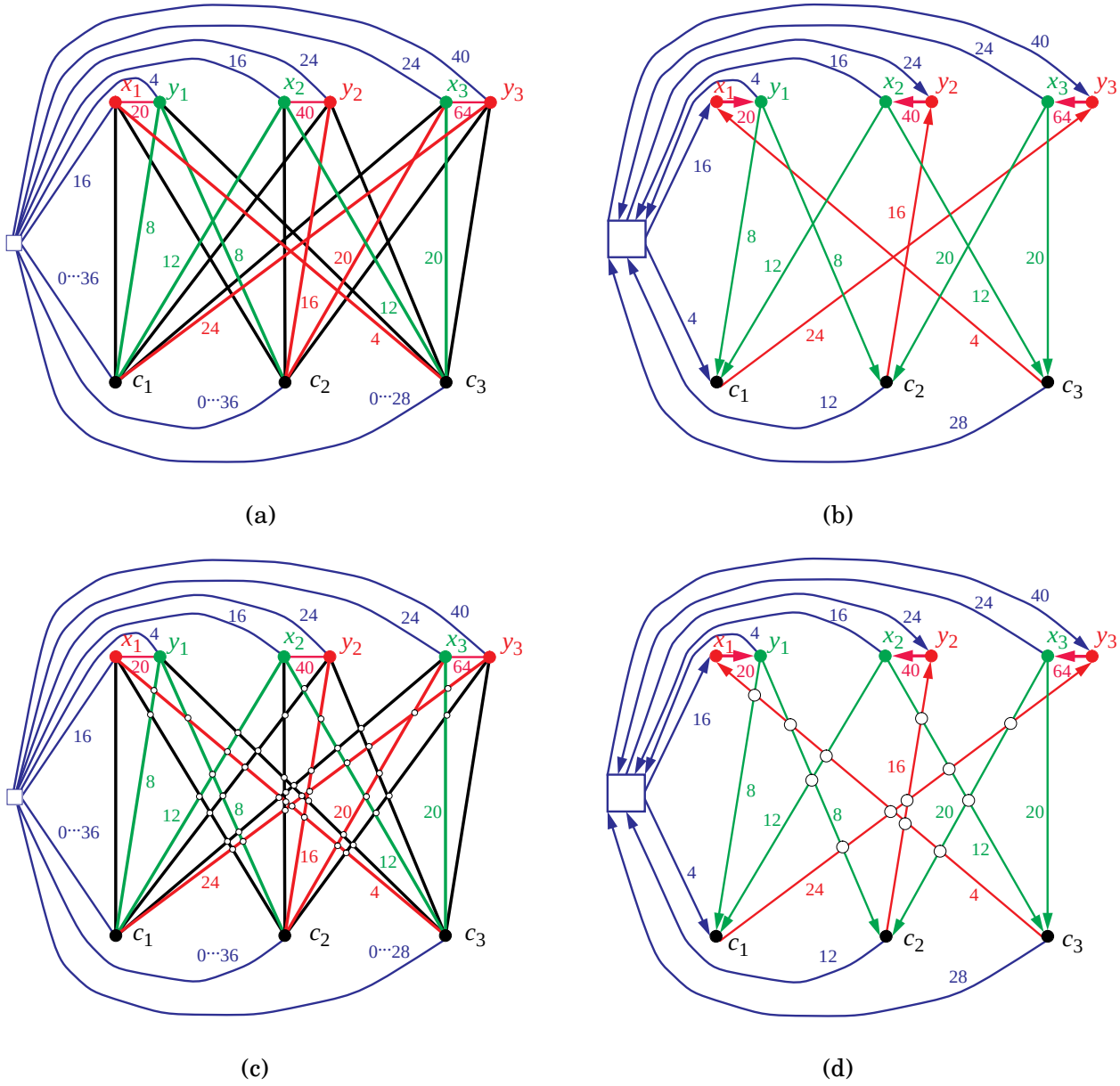


**Figure 17:** **(a)** Switch-flow network $\mathcal{N}$ with parameter $\theta = 4$ associated with the the NOT-ALL-EQUAL-3-SAT instance $\mathcal{S}$ with clauses $c_1 = y_1x_2y_3$, $c_2 = y_1y_2x_3$, and $c_3 = x_1x_2x_3$. The clause edges with zero capacity range are shown with black lines. **(b)** Feasible flow for $\mathcal{N}$ corresponding to the satisfying truth assignment $(y_1, x_2, x_3)$ for $\mathcal{S}$. Only the edges with nonzero flow are shown. **(c)** Planar switch-flow network $\mathcal{P}$ associated with $\mathcal{S}$. The edges with zero capacity range are shown with black lines. **(d)** Feasible flow for $\mathcal{P}$ corresponding to the satisfying truth assignment $(y_1, x_2, x_3)$ for $\mathcal{S}$. Only the edges with nonzero flow are shown.

all incoming or all outgoing because of the conservation property at vertex $c_j$ and the choice of capacity range for the dummy edge incident on $c_j$. We obtain:

**Lemma 6** [17] *An instance $\mathcal{S}$ of* NOT-ALL-EQUAL-3-SAT *is satisfiable if and only if the associated switch-flow network $\mathcal{N}$ admits a feasible flow. Also, given a feasible flow for $\mathcal{N}$, a satisfying truth assignment for $\mathcal{S}$ can be computed in time $O(nm)$, where $n$ and $m$ are the number of variables and clauses of $\mathcal{S}$, respectively.*

Now, starting from $\mathcal{N}$, we construct a planar switch-flow network $\mathcal{P}$ consisting of $O(n^2m^2)$ vertices in $O(n^2m^2)$ time. Network $\mathcal{P}$ has the property that network $\mathcal{N}$ admits a feasible flow if and only if network $\mathcal{P}$ admits a feasible flow, and a feasible flow for $\mathcal{N}$ can be computed from a feasible flow for $\mathcal{P}$ in $O(n^2m^2)$ time.

$\mathcal{P}$ is constructed as follows (see Fig. 17). First, we construct a drawing of $\mathcal{N}$ such that the literal vertices and the clause vertices are arranged on two parallel lines, and crossings occur only between clause edges. Next, we replace the crossings formed by the clause edges with new vertices, called *crossing* vertices. We call *fragment edges* the edges originated by the splitting of the clause edges. Each fragment edge inherits the capacity range of the originating clause edge.

Using Lemma 6, we obtain the main result of this section.

**Theorem 18** [17] *Given an instance $\mathcal{S}$ of* NOT-ALL-EQUAL-3-SAT *with $n$ variables and $m$ clauses, the associated planar switch-flow network $\mathcal{P}$ has $O(n^2m^2)$ vertices and edges, and can be constructed in $O(n^2m^2)$ time. Instance $\mathcal{S}$ is satisfiable if and only if network $\mathcal{P}$ admits a feasible flow. Also, given a feasible flow for $\mathcal{P}$, a satisfying truth assignment for $\mathcal{S}$ can be computed in time $O(n^2m^2)$. Moreover, if $n \geq 3$ and $m \geq 3$, then $\mathcal{P}$ is triconnected.*

## 5.3 Upward Planarity Testing

In this section we show how to reduce the problem of computing a feasible flow in the planar switch-flow network $\mathcal{P}$ associated with a NOT-ALL-EQUAL-3-SAT instance $\mathcal{S}$ to the problem of testing the upward planarity of a suitable digraph. We set parameter $\theta$ equal to 4.

Now, we construct on orientation $\vec{\mathcal{P}}$ of $\mathcal{P}$ as follows (see Fig. 18):

- Every literal edge $(x_i, y_i)$ is oriented from $x_i$ to $y_i$.
- Every fragment edge is oriented away from the clause vertex and towards the literal vertex.
- Every dummy edge incident on a literal vertex is oriented towards the dummy vertex, and every dummy edge incident on a clause vertex is oriented towards the clause vertex.

By Theorem 18, if $\mathcal{S}$ has at least three clauses and variables each, the planar embedding of $\mathcal{P}$ and the dual graph $\mathcal{D}$ of $\mathcal{P}$ are unique. We construct the dual digraph $\vec{\mathcal{D}}$ of $\vec{\mathcal{P}}$ by orienting every dual edge of $\mathcal{D}$ from the face on the left to the face on the right of the primal edge (see Fig. 18).

**Lemma 7** [17] *The dual digraph $\vec{\mathcal{D}}$ of $\vec{\mathcal{P}}$ is upward planar, triconnected, acyclic and has exactly one source and one sink, denoted with $s$ and $t$.*

Starting from digraph $\vec{\mathcal{D}}$ we construct a new digraph $\vec{\mathcal{G}}$ by replacing the edges of $\vec{\mathcal{D}}$ with subgraphs (tendrils or wiggles), as follows (see Fig. 19):

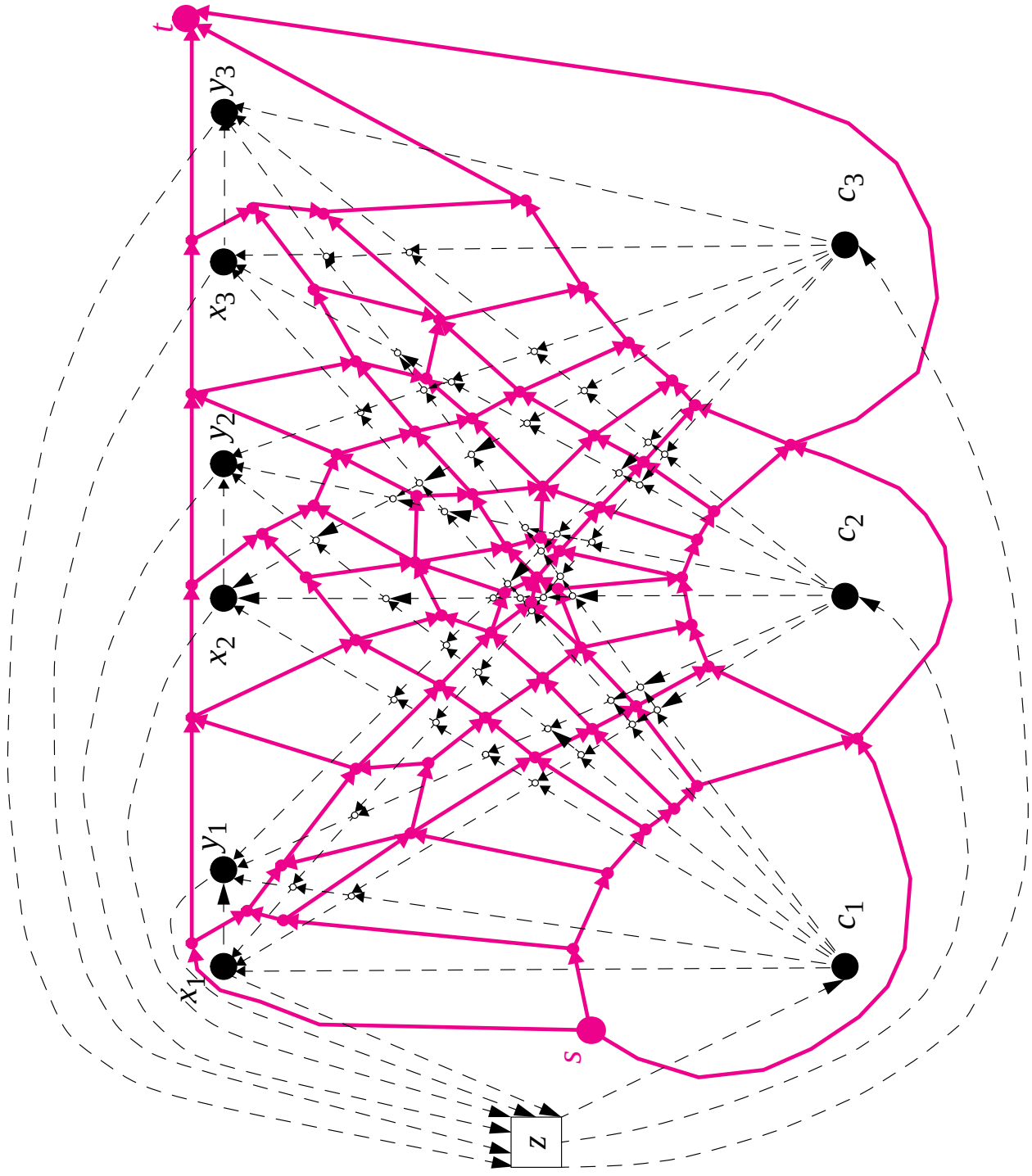


**Figure 18:** Orientation $\vec{\mathcal{P}}$ (drawn with dashed lines) of the network $\mathcal{P}$ shown in Fig. 17(c) and dual digraph $\vec{\mathcal{D}}$ (drawn with solid lines) of $\vec{\mathcal{P}}$.

- Every edge of $\vec{\mathcal{D}}$ that is the dual of a literal edge, fragment edge, or dummy edge incident on a literal vertex is replaced with tendril $T_c$, where $[c]$ is the capacity range of the dual edge. Note that $c$ is a multiple of parameter $\theta$.
- Every edge of $\vec{\mathcal{D}}$ that is the dual of a dummy edge incident on a clause vertex is replaced with wiggle $W_c$, where $[0 \cdots c]$ is the capacity range of the dual edge.

The vertices of $\vec{\mathcal{G}}$ that are also in $\vec{\mathcal{D}}$ are called *primary vertices*. The remaining vertices of $\vec{\mathcal{G}}$ are called *secondary vertices*. By Lemma 7 and the construction of digraph $\vec{\mathcal{G}}$, all the embeddings of $\vec{\mathcal{G}}$ are obtained by choosing one of the two possible flips for each tendril.

**Lemma 8** [**17**] *Digraph $\vec{\mathcal{G}}$ is upward planar if and only if the tendrils can be flipped and the wiggles can be arranged such that for every face the total contribution of the tendrils and wiggles is zero.*

The proof of Lemma 8 uses the result of Theorem 2.

We establish the following correspondence between digraph $\vec{\mathcal{G}}$ and network $\mathcal{P}$ (see Fig. 19): the faces of $\vec{\mathcal{G}}$ correspond to the vertices of $\mathcal{P}$; the tendrils and wiggles of $\vec{\mathcal{G}}$ correspond to the the edges of $\mathcal{P}$; flipping a tendril of $\vec{\mathcal{G}}$ corresponds to orienting an edge of $\mathcal{P}$; the contribution of a tendril or wiggle of $\vec{\mathcal{G}}$ corresponds to the flow in an edge of $\mathcal{P}$; the balance of the contributions of the tendrils and wiggles in the faces of $\vec{\mathcal{G}}$ corresponds to the conservation of flow at the vertices of $\mathcal{P}$.

**Theorem 19** [**17**] *Given an instance $\mathcal{S}$ of* NOT-ALL-EQUAL-3-SAT *with $n$ variables and $m$ clauses and the associated planar switch-flow network $\mathcal{P}$, digraph $\vec{\mathcal{G}}$ associated with $\mathcal{S}$ and $\mathcal{P}$ has $O(n^3m^2)$ vertices and edges, and can be constructed in $O(n^3m^2)$ time. Instance $\mathcal{S}$ is satisfiable and network $\mathcal{P}$ admits a feasible flow if and only if digraph $\vec{\mathcal{G}}$ is upward planar. Also, given an upward planar embedding for $\vec{\mathcal{G}}$, a feasible flow for $\mathcal{P}$ and a satisfying truth assignment for $\mathcal{S}$ can be computed in time $O(n^3m^2)$.*

From Theorems 11, 18 and 19 we conclude:

**Corollary 1** [**17**] *Upward planarity testing is NP-complete.*

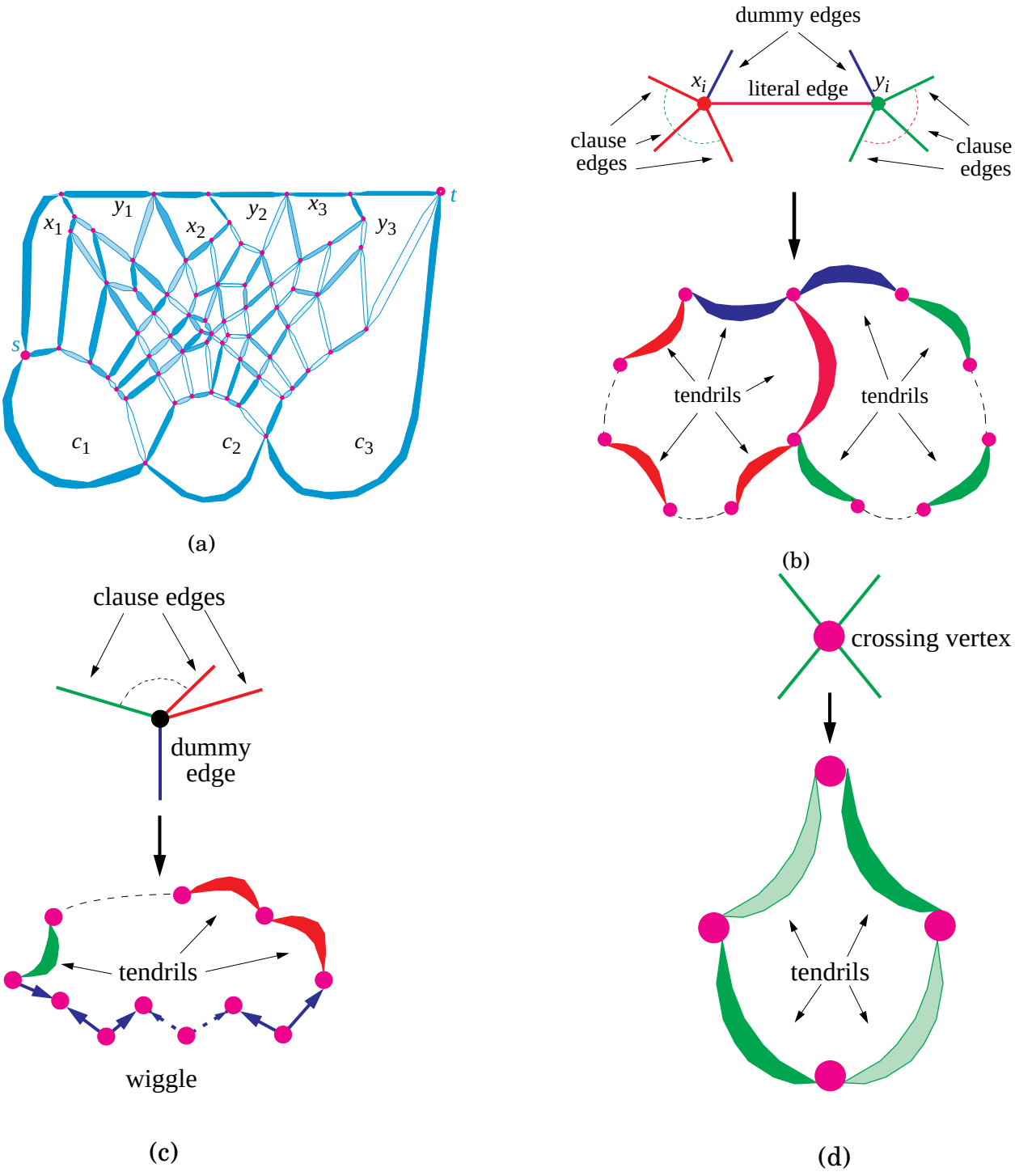


**Figure 19:** Schematic illustration of: **(a)** digraph $\vec{\mathcal{G}}$ obtained from $\vec{\mathcal{D}}$ by replacing edges with tendrils and wiggles; **(b)** the two faces of $\vec{\mathcal{G}}$ associated with literal vertices $x_i$ and $y_i$ of $\mathcal{P}$; **(c)** the face of $\vec{\mathcal{G}}$ associated with a clause vertex of $\mathcal{P}$; and **(d)** the face of $\vec{\mathcal{G}}$ associated with a crossing vertex of $\mathcal{P}$.

# References


[1] K. A. Baker, P. Fishburn, and F. S. Roberts. Partial orders of dimension 2. *Networks*, 2:11–28, 1971.

[2] P. Bertolazzi, R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis. How to draw a series-parallel digraph. *Internat. J. Comput. Geom. Appl.*, 4:385–402, 1994.

[3] P. Bertolazzi and G. Di Battista. On upward drawing testing of triconnected digraphs. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 272–280, 1991.

[4] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12:476–497, 1994.

[5] P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. In *1st Annual European Symposium on Algorithms (ESA '93)*, volume 726 of *Lecture Notes in Computer Science*, pages 37–48. Springer-Verlag, 1993.

[6] G. Birkhoff. *Lattice Theory*. American Mathematical Society, Providence, RI, 1967.

[7] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. North-Holland, New York, NY, 1976.

[8] K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13:335–379, 1976.

[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Mass., 1990.

[10] H. de Fraysseix and P. Rosenstiehl. A depth-first-search characterization of planarity. *Annals of Discrete Mathematics*, 13:75–80, 1982.

[11] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4:235–282, 1994.

[12] G. Di Battista, W. P. Liu, and I. Rival. Bipartite graphs, upward drawings, and planarity. *Inform. Process. Lett.*, 36:317–322, 1990.

[13] G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoret. Comput. Sci.*, 61:175–198, 1988.

[14] G. Di Battista and R. Tamassia. On-line graph algorithms with SPQR-trees. In *Automata, Languages and Programming (Proc. 17th ICALP)*, volume 442 of *Lecture Notes in Computer Science*, pages 598–611, 1990.

[15] G. Di Battista, R. Tamassia, and I. G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discrete Comput. Geom.*, 7:381–401, 1992.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.

[17] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 286–297. Springer-Verlag, 1995.

[18] J. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2:135–158, 1973.

[19] J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[20] M. D. Hutton and A. Lubiw. Upward planar drawing of single source acyclic digraphs. In *Proc. 2nd ACM-SIAM Sympos. Discrete Algorithms*, pages 203–211, 1991.

[21] D. Kelly. Fundamentals of planar ordered sets. *Discrete Math.*, 63:197–216, 1987.

[22] D. Kelly and I. Rival. Planar lattices. *Canad. J. Math.*, 27(3):636–665, 1975.

[23] A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In *Theory of Graphs: Internat. Symposium (Rome 1966)*, pages 215–232, New York, 1967. Gordon and Breach.

[24] A. Papakostas. Upward planarity testing of outerplanar dags. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 298–306. Springer-Verlag, 1995.

[25] C. Platt. Planar lattices and planar graphs. *J. Combin. Theory Ser. B*, 21:30–39, 1976.

[26] I. Rival. Reading, drawing, and order. In I. G. Rosenberg and G. Sabidussi, editors, *Algebras and Orders*, pages 359–404. Kluwer Academic Publishers, 1993.

[27] C. Thomassen. Planar acyclic oriented graphs. *Order*, 5(4):349–361, 1989.