

# Upward Planar Drawing of Single Source Acyclic Digraphs

by

Michael D. Hutton

A thesis  
presented to the University of Waterloo  
in fulfilment of the  
thesis requirement for the degree of  
Master of Mathematics  
in  
Computer Science

Waterloo, Ontario, Canada, 1990

©Michael D. Hutton 1990



I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

## **Acknowledgements**

The results of this thesis arise from joint research with my supervisor, Anna Lubiw. Anna also deserves credit for my initial interest in Algorithms and Graph Theory, for her extra effort to help me make my deadlines, and for financial support from her NSERC Operating Grant.

Thanks also to NSERC for financial support, and to my readers Kelly Booth and Derick Wood for their valuable comments and suggestions.

## **Dedication**

This thesis is dedicated to my parents, David and Barbara on the occasion of their 26'th wedding anniversary. Without their years of hard work, love, support and endless encouragement, it would not exist.

Michael David Hutton

August 29, 1990

## Abstract

A upward plane drawing of a directed acyclic graph is a straight line drawing in the Euclidean plane such that all directed arcs point upwards. Thomassen [30] has given a non-algorithmic, graph-theoretic characterization of those directed graphs with a single source that admit an upward drawing. We present an efficient algorithm to test whether a given single-source acyclic digraph has a plane upward drawing and, if so, to find a representation of one such drawing.

The algorithm decomposes the graph into biconnected and triconnected components, and defines conditions for merging the components into an upward drawing of the original graph. For the triconnected components we provide a linear algorithm to test whether a given plane representation admits an upward drawing with the same faces and outer face, which also gives a simpler (and algorithmic) proof of Thomassen's result. The entire testing algorithm (for general single source directed acyclic graphs) operates in  $O(n^2)$  time and  $O(n)$  space.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Preliminaries</b>	<b>5</b>
<b>3</b>	<b>Related Results</b>	<b>10</b>
3.1	Testing for Planarity of Graphs . . . . .	11
3.1.1	The Hopcroft-Tarjan Algorithm . . . . .	11
3.1.2	The Lempel-Even-Cederbaum Algorithm . . . . .	12
3.2	Drawing Planar Graphs . . . . .	15
3.3	Testing for Upward Planarity . . . . .	16
3.4	Drawing Upward Planar Graphs . . . . .	18
3.5	Graphs and Partially Ordered Sets . . . . .	19
<b>4</b>	<b>Strongly-Equivalent Upward Planarity</b>	<b>22</b>
4.1	Implementation Details . . . . .	25
<b>5</b>	<b>Separation into Bi-Connected Components</b>	<b>28</b>

<b>6</b>	<b>Separation into Tri-Connected Components</b>	<b>30</b>
6.1	Some Basic Results . . . . .	32
6.1.1	Facial Structure of Upward Planar Graphs . . . . .	32
6.1.2	Properties of Cut Sets in Digraphs . . . . .	33
6.1.3	Topological Properties of Upward Planar Graphs . . . . .	36
6.2	Marker Components . . . . .	39
6.3	The cut-set $\{u, v\}$ , where $u$ and $v$ are incomparable . . . . .	40
6.4	The cut-set $\{u, v\}$ , where $u < v$ , $u \neq s$ . . . . .	43
6.5	The cut-set $\{s, v\}$ . . . . .	48
<b>7</b>	<b>An Algorithm for Upward Planar Embedding</b>	<b>54</b>
<b>8</b>	<b>Drawing Upward Planar Graphs</b>	<b>59</b>
<b>9</b>	<b>Conclusions</b>	<b>60</b>

# List of Figures

2.1	Upward planar and non-upward planar graphs. . . . .	9
3.1	$st$ -numberings of $G$ (a) and $G_4$ (b); bush-form $B_4$ for $G_4$ (c). . . . .	14
3.2	Bush form (a) and its associated PQ-tree (b). . . . .	15
4.1	Cases for $v$ in Algorithm S-E-UP-PLANAR. . . . .	27
5.1	Construction of $\overline{G}$ from its bi-connected components . . . . .	29
6.1	Added complication of 2-vertex cut-sets. . . . .	31
6.2	Facial structure in an upward plane drawing. . . . .	33
6.3	Transformations preserving upward planarity. . . . .	38
6.4	Marker Graphs. . . . .	40
6.5	Merging of $S$ and $H$ when cut-set $\{u, v\}$ is incomparable. . . . .	42
6.6	Merging of $S$ and $H$ with cut-set $\{u, v\}$ and $u < v$ . . . . .	46

# Chapter 1

## Introduction

There are a wide range of results dealing with drawing, representing, or testing planarity of graphs. Fáry [16] showed that every planar graph can be drawn in the plane using only straight lines for the edges. Tutte [33] showed that every 3-connected planar graph admits a convex straight-line drawing, where the facial cycles other than the unbounded face are all convex polygons. The first linear time algorithm for testing planarity of a graph was given by Hopcroft and Tarjan [18]. Planar graph layout has many interesting applications in automated circuit design, for representation of network flow problems (e.g. PERT graphs in software engineering [14]) and artificial intelligence.

A *upward plane drawing* of an directed acyclic graph (DAG) is a straight line drawing in the Euclidean plane such that all directed arcs point upwards (i.e. from a lower to a higher  $y$ -coordinate). A DAG is *upward planar* if it has an upward plane drawing. Kelly [20] and Kelly and Rival [21] have shown that every upward plane drawing with monotonic edges (each arc is drawn as a curve monotone increasing in the  $y$ -direction) has a similar upward plane straight line drawing, an analogue

of Fáry’s result for general planar graphs. Thus, the straight line condition of the upward planar definition does not restrict the class of *upward planar* graphs. We will use this fact to simplify the discussion. The definition of *upward planar* will be detailed in Chapter 2. In this thesis we will consider only single source DAGs. Initially we will be concerned with constructing an abstract embedding. This will then be extended to encompass a physical drawing algorithm. The difference between these terms will be explained in Chapter 2. Our main result is an  $O(n^2)$  algorithm to test whether a given graph is upward planar, and if so, to give a representation of an upward plane drawing for it. This result is based on a graph-theoretic result of Thomassen [30, Theorem 5.1]:

**Theorem 1.1 (Thomassen)** *Let  $\Gamma$  be a planar representation of a single source DAG  $G$ . Then there exists an upward plane drawing  $\Gamma'$  strongly equivalent to (i.e. having the same faces as)  $\Gamma$  if and only if the source  $\alpha$  of  $G$  is on the outer face of  $\Gamma$ , and for every cycle  $\Sigma$  in  $\Gamma$ ,  $\Sigma$  has a vertex  $\beta$  which has a directed edge from each of its two neighbours on  $\Sigma$ , and which is not the tail of any directed edge inside  $\Sigma$ .*

The terms of this theorem are defined more precisely in Chapter 2. Intuitively, for a graph  $G$  with planar embedding  $\Gamma$ , if some cycle  $\Sigma$  of  $\Gamma$  has every vertex which is on a directed edge from each of its two neighbours also having another edge directed “into”  $\Sigma$ , then no vertex on  $\Sigma$  can be the maximal vertex of  $\Sigma$ , so no drawing  $\Gamma'$  with the same faces as  $\Gamma$  (i.e. containing  $\Sigma$  as a cycle) can be an upward plane drawing of  $G$ . So intuitively, Thomassen’s condition is necessary. The interesting part of the theorem is that the condition is also sufficient for  $\Gamma$  to be strongly equivalent to an upward planar representation of  $G$ .

Thomassen notes that a 3-connected graph has a unique planar embedding (up to the choice of the outer face) and concludes that his theorem provides a “good

characterization” of 3-connected upward planar graphs. Neither a concrete time bound nor an efficient algorithm is, however, given, nor is a method for isolating and joining 3-connected components.

The problem thus decomposes into two main issues. The first is to describe Thomassen’s result algorithmically; we do this in Chapter 4 with a linear time algorithm, at the same time providing an alternative proof of his theorem. The second is to isolate the triconnected components of the input graph, and determine how to put the “pieces” back together after the embedding of each is complete. This more complex issue comprises the remainder of the thesis.

The algorithm for splitting the input into triconnected components and merging the embeddings of each operates in  $O(n^2)$  time. Since a triconnected graph is uniquely embeddable in the plane, up to the choice of the outer face, and the number of possible external faces of a planar graph is linear by Euler’s formula, the overall time to test a given 3-connected component is also  $O(n^2)$ , so the entire algorithm is quadratic.

The primary focus of this thesis is to test a single-source DAG for upward planarity, and to generate an upward planar embedding in the process (whenever possible). This process can be extended to a drawing algorithm by transforming the embedding for use by the known algorithm of DiBattista and Tamassia [10] to draw single-source, single-sink planar DAGs. The main issue in this extension is the size of the output of the drawing algorithm, which we will address in Chapter 3 and again in Chapter 7. Upward plane drawings can require an exponential sized coordinate grid (a precise definition is given in Chapter 3), so no algorithm can, in general, output an upward plane drawing in time polynomial in the size of the graph.

Chapter 4 discusses an algorithm for determining an upward planar representation for a 3-connected graph with a prescribed outer face. Chapters 5 and 6 discuss the isolation of the components of an input graph  $G$ , and merging of the (recursively determined) components into an upward planar embedding. The combined testing/embedding algorithm is given in Chapter 7. The “abstract embedding” output by the algorithm of Chapter 7 is a representation of a particular member of the *set* of possible drawings using the cyclic order of edges around each vertex, and some additional structural information, to identify the faces; this will be defined more precisely in Chapter 2. Chapter 8 extends the previous result to the construction of a physical drawing from the abstract embedding of Chapter 7 (i.e. giving Euclidean coordinates for the vertices).

# Chapter 2

## Preliminaries

For the most part we use the terminology and notation of Bondy and Murty [3], supplemented by that of Thomassen [30].

A (combinatorial) *graph*  $G$  consists of a set of *vertices* or nodes  $V$  and a set  $E$  of pairs of vertices called *edges* connecting these vertices. Late lower case Roman letters (e.g.  $u, v, w, x, y$ ) denote vertices unless otherwise stated, and early letters (e.g.  $a, b, c, d, e$ ) denote edges. A graph is *directed*, called a *digraph*, if each edge  $e = (u, v)$  is *directed* from the vertex  $u$ , called the *tail* of  $e$ , to the vertex  $v$ , called the *head* of  $e$ , also denoted  $u \rightarrow v$ . A *walk*  $W = v_0e_1v_1e_2v_2 \dots e_kv_k$  of  $G$  is an alternating sequence of (possibly repeating) incident vertices and edges of  $G$ . A *path* in  $G$  is an alternating sequence of distinct (except possibly for the first and last vertex) incident vertices and edges; a path is hence a walk with no repeating edges or vertices, other than possibly  $v_0$  and  $v_n$ . A *cycle*  $C$  of a graph  $G$  is a exactly such a *closed* path (from one node back to itself), denoted  $u \overset{\pm}{\rightarrow} u$ . Analogously in a directed graph, a *directed path* is also denoted by  $u \overset{*}{\rightarrow} v$ , (connected through 0 or more edges) or  $u \overset{\pm}{\rightarrow} v$  (connected by 1 or more edges) or  $u \overset{k}{\rightarrow} v$  (connected by exactly  $k$  edges).

A directed path  $u \xrightarrow{+} u$  is called a *directed cycle*. When the notation is ambiguous, we specifically state whether a path is directed or undirected. A directed graph is *acyclic* if it contains no directed cycles (although the underlying undirected graph may contain cycles). A *subgraph*  $H$  of a graph  $G$  is a subset of the nodes and edges of  $G$ . Subgraph  $H$  is *proper* if it is not equal to  $G$ .  $G[H]$  is the vertex-induced subgraph of  $G$  — the vertices of  $H \subseteq G$ , and any edges of  $G$  connecting nodes in  $H$ .

A *planar representation* of a graph  $G$ , denoted by an upper case Greek letter such as  $\Gamma$  or by  $\overline{G}$ , is an abstract embedding (in terms of a data structure) in the plane. We will describe such an embedding in terms of a cyclic (counter-clockwise) ordering of edges/vertices around each vertex in the embedding, and a specification of which face is the outer face. An *upward planar representation* is such an embedding, with an additional specification of the extra edges required to make it have no sinks other than on the outer face—to *resolve* the internal sinks. This allows it to be extended to a single-source, single-sink graph which can be drawn easily. A *plane drawing* is a physical realization of a planar embedding; where each vertex is assigned a coordinate point in the Euclidean plane, and each edge is an curve or arc between points in the plane; no two arcs intersect except at incident vertices on their endpoints.  $G$  is *planar* if it can be represented by a plane drawing  $\Gamma$ . As mentioned in Chapter 1, if  $G$  has an upward plane drawing with monotonic arcs as edges, it also has some upward plane drawing with straight line arcs as edges. The latter is the definition we use for upward plane, but as a result of this equivalence we will ignore the straight line issue for the purposes of testing upward planarity, and address it as required for the upward drawing in Chapter 8. In a planar representation, the representation of a cycle which has no vertices inside of it is called a *face* or *facial cycle*, and the outermost face (having no vertices

outside of it, and all vertices of  $G$  inside or on it) is called the *outer* facial cycle. Two representations of a graph  $G$  are *equivalent* if they have identical facial cycles, and *strongly equivalent* if they are equivalent and have the same face identified as the outer face.

We say that a drawing of  $G$  is *oriented* if a direction (in the Euclidian plane) is imposed upon the directed arcs representing the edges of  $G$ . For our purposes, this means that all directed arcs must be monotonic upward (positive in the  $y$  direction), i.e. *upward plane*. Unless otherwise stated, all figures in the thesis are drawn using this convention, so arrow heads will often be omitted from the graph edges. A representation  $\Gamma$  of  $G$  is *convex* if all facial cycles of  $\Gamma$  are convex polygons.

A digraph  $G$  is *connected* if there exists an undirected path between any two vertices.  $G \setminus v \equiv G[V - \{v\}]$  denotes  $G$  with  $v$  and all edges incident to  $v$  in  $G$  removed.  $G$  is  $k$ -connected if the removal of at least  $k$  vertices is required to *disconnect* the graph. An equivalent definition (by Menger's Theorem [3]) would be to say  $G$  is  $k$ -connected if and only if there exist  $k$  vertex-disjoint undirected paths between any two vertices. A set of vertices whose removal disconnects the graph is a *cut-set*. A *cut vertex* and *separation pair* denote cut-sets of size one and two respectively. A graph which has no cut vertex is *biconnected* (2-connected). A graph with no separation pair is *triconnected* (3-connected). For  $G$  with cut vertex  $v$ , a *component* of  $G$  with respect to  $v$  is formed from a connected component  $H$  of  $G \setminus v$  by adding to  $H$  the vertex  $v$  and all edges between  $v$  and  $H$ . For  $G$  with separation pair  $\{u, v\}$ , a *component* of  $G$  with respect to  $\{u, v\}$  is formed from a connected component  $H$  of  $G \setminus \{u, v\} \equiv G[V - \{u, v\}]$  by adding to  $H$ ,  $u, v$  and all edges between  $u, v$  and vertices of  $H$ . The edge  $(u, v)$ , if it exists, forms a component by itself. If  $S$  is a cut-set, then the *vertices of attachment* of a component  $H$  of  $G \setminus S$  are those vertices in  $H$  which share an edge with some vertex of  $S$  in the graph

$G$ . An algorithm for finding tri-connected components<sup>1</sup> in linear time is given in Hopcroft and Tarjan [19]. A related concept is that of *graph union*. We define  $G_1 \cup G_2$ , for components with “shared” vertices to be the *inclusive* union of all vertices and edges. That is, for  $v$  in both  $G_1$  and  $G_2$ , the vertex  $v$  in  $G_1 \cup G_2$  is adjacent to edges in each of the subgraphs  $G_1$  and  $G_2$ .

Contracting an edge  $e = (u, v)$ , denoted  $G/e$ , results in a graph  $G'$  with the edge  $e$  removed, and vertices  $u$  and  $v$  *identified* (as the same vertex in  $G'$ ). Inserting new vertices within edges of  $G$  generates a *subdivision* of  $G$ . A *directed subdivision* (of a digraph) is a subdivision within directed edges (i.e.  $(u, v)$  can be subdivided into  $(u, w), (w, v)$ , but not  $(u, w), (v, w)$ ).  $G_1$  and  $G_2$  are *homeomorphic* if both are subdivisions of some other graph  $G$ .  $G$  is planar if and only if every subdivision of  $G$  is planar [3].

In a directed graph, the *in-degree* of a vertex  $v$  is the number of edges directed towards  $v$ , denoted  $\text{deg}^-v$ . Analogously the *out-degree* ( $\text{deg}^+v$ ) of  $v$  is the number of edges directed away from  $v$ . A vertex of in-degree 0 is a *source* in  $G$ , and a vertex of out-degree 0 is a *sink*.

Directed/oriented graphs are often used to represent partially ordered sets (*posets*); in particular, a *corresponding digraph*  $G$  of a poset  $P$  has an directed path  $u \xrightarrow{+} v$  in  $G$  if and only if  $u < v$  in  $P$ . If  $G$  has no transitive edges (i.e.  $(u, v) \notin G$  if  $\exists w|(u, w)$  and  $(w, v) \in G$ ), then the drawing of  $G$  is called a *Hasse diagram*. The problem of drawing plane Hasse diagrams is equivalent to the problem of drawing general DAGs, since subdividing every edge of a DAG produces a corresponding graph with no transitive edges.

Adopting some poset notation: two vertices  $u, v$  are *comparable* if there is a

---

<sup>1</sup>Note that Hopcroft and Tarjan’s “components” include an extra  $(u, v)$  edge.

directed path  $u \xrightarrow{*} v$  ( $u < v$ ) or  $v \xrightarrow{*} u$  ( $v < u$ ), and *incomparable* otherwise. We often use  $u < v$  to represent the existence of a directed path from  $u$  to  $v$ . If  $(u, v)$  is an edge in the digraph of  $P$ , then  $u$  *dominates*  $v$ .

Figure 2.1 shows two drawings of upward planar and non-upward planar graphs. The first is a drawing of an upward planar graph. The second is planar, but non-upward planar, since placing  $v$  inside the face  $f$  would eliminate crossing, but require a downward edge. Recall our convention that edges are directed upward unless otherwise stated.

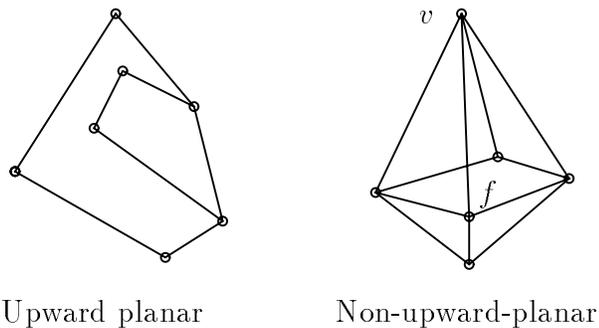


Figure 2.1: Upward planar and non-upward planar graphs.

# Chapter 3

## Related Results

The problem of drawing a graph on a two dimensional surface has many different applications. The most obvious are in circuit design/layout, and VLSI, especially in the area of automated tools to realize schematics of an abstract circuit. Other applications are to software engineering and information analysis (PERT diagrams, subroutine-call charts, data-flow diagrams); databases (entity relationship graphs); and knowledge representation (ISA hierarchies and neural networks).

The (algorithmic) problem of graph drawing was brought into the open after two papers by Tutte, on convex (and hence straight-line, except possibly for the edges on the outer face) drawings of graphs [33], and graph drawing in general [34]. Continuing Tutte's results on convex graphs, Thomassen [31] has presented inductive criteria for the existence of a convex representation of a graph in the plane. This was implemented as a linear time algorithm by Chiba et al [5]. These are significant to the topic at hand because they provide much of the graph-theoretical background evident in the works which followed.

For general graphs, the problem of a “good” presentation is neither solved nor

even clearly defined. Since no obvious set of concrete requirements is known for drawing a graph to easily portray its meaning to the (human) reader, work thus far has been in exploring various arbitrary aesthetics: minimum line crossings; minimum area on some grid or set of points; preference for polygonal lines, minimum bends within these, or straight lines; curves which are monotonic in some direction; and isolation and display of symmetry in automorphisms or near automorphisms. Many of these aesthetics have either been proven NP-hard to determine or display, or their complexity remains open. Some aesthetics are conflicting when applied in tandem.

Of especial interest is the sub-area of drawing *planar* graphs. For schematics it is often necessary to have a planar realization of a circuit, as wires cannot cross. For other applications, it is desirable to present a graph as planar whenever possible. Thus algorithms for testing planarity, and operating upon planar graphs have been heavily studied.

For a (continuing) bibliographic survey of the field of graph drawing, including a list of aesthetics (“graphic standards”) and complexity results on their representation, see Tomassia and Eades’ “Annotated Bibliography” [29].

## 3.1 Testing for Planarity of Graphs

### 3.1.1 The Hopcroft-Tarjan Algorithm

The first linear time planarity testing algorithm is due to Hopcroft and Tarjan [18]. This algorithm first isolates the biconnected components of the graph, which are easy to put together after each is separately tested.

The operation on the biconnected components is a *path addition* algorithm: A depth first search is performed to order the graph into a tree-like structure with *tree-edges* and *non-tree-edges* or *fronds*—called a *palm tree*. A simple  $k$ -circuit is chosen to contain  $k - 1$  tree edges, and one frond. The set of paths generated by the deletion of the cycle from  $G$  are then added to the cycle, one path at a time, generating  $G$ .

The paths have to be added either on the left or right of the tree-edge path of the  $k$ -circuit, so two stacks are used to record the relevant information, and components are flipped back and forth between the two stacks (a non-trivial process) as added paths require. This process continues until all paths are added or a path cannot be added.

### 3.1.2 The Lempel-Even-Cederbaum Algorithm

One competing planarity algorithm is a *vertex addition* algorithm due to Lempel, Even and Cederbaum [22]. The original version required  $O(|V| \cdot |E|)$  time. This is exactly  $O(n^2)$  time, under the observation that a planar graph has no more than a linear number of edges (Euler's formula, [3]), and the further assumption that the graph is connected (requiring at least  $n - 1$  edges). The algorithm can, however, be modified to run in linear time, and is more attractive to our discussion because it also has application to the drawing of *upward* planar graphs.

The Lempel, Even and Cederbaum algorithm is divided into two main phases. The first is to compute an *st-numbering* of the graph, which can be realized in linear time due to a result of Even and Tarjan [15]. The second phase applies the *st-numbering* to the vertex-addition algorithm, using the PQ-tree data structure of Booth and Lueker [4] to achieve the embedding in linear time.

**Definition 3.1** An *st-numbering* is a numbering  $1 \dots n$  of the vertices of a biconnected graph  $G$  such that two distinguished vertices  $s$  and  $t$  are numbered 1 and  $n$  respectively, and for any other vertex  $j$ , there exist adjacent (to  $j$ ) vertices,  $i$  and  $k$ , such that  $i < j < k$ .

Such an ordering can be imposed on any biconnected undirected graph (Lempel, Even and Cederbaum) [22] and can be found in linear time (Even and Tarjan) [15].

We notice that an *st-numbering* of a graph  $G$  identifies  $s$  as the single source of  $G$ , and  $t$  as the single sink of  $G$  (directing edges from smaller to bigger numbers). In fact, the following is a more general indication of its usefulness:

**Lemma 3.2** [Lempel, Even and Cederbaum] *Let  $1 \leq k \leq n$ . If edge  $(s, t)$  is drawn on the boundary of the outer face in an embedding  $\overline{G}$  of  $G$ , then all the vertices and edges of  $G - G_k$  are drawn in the outer face of the plane subgraph  $\overline{G_k}$  of  $G$ , where  $G_k$  is the subgraph induced by the vertices  $1 \dots k$  of the *st-numbering* of  $G$ .*

We can use this ordering to effect an embedding of  $G$ : Define *virtual edges* as edges between vertices of  $G_k$  and vertices of  $G - G_k$ , and *virtual vertices* as endpoint vertices of virtual edges in  $G - G_k$ . The virtual vertices are labelled as their corresponding points in  $G - G_k$ , but kept separate. For example, Figure 3.1 shows a graph  $G$  labeled with an *st-numbering*, the subgraph  $G_4$  of  $G$ , and a bush form (to be defined shortly) which illustrates the (multiple) virtual vertices for the actual vertices numbered 5 and 6 in the bush form for  $G_4$ . Notice that the problem of finding a planar embedding of  $G_6$  consists of mapping all sets of identically labeled virtual vertices (sets  $\{5_i\}$  and  $\{6_i\}$  in the example) of the bush form of  $G_4$  to one single point (points 5 and 6 in the example) in the embedding. Thus to find an embedding of  $G$ , we first find a planar embedding of some  $G_l$  ( $l < k$ ). By the

Lempel-Even-Cederbaum theorem (above), all vertices and edges of  $G_k - G_l$  are in the outer face of the embedding  $\overline{G_l}$ , so the general step reduces to finding such a mapping of the corresponding virtual vertices  $\{i\}$  to a common actual vertex  $i$ . This is accomplished using a data structure called a PQ-tree.

**Definition 3.3** A PQ-tree is a data structure with three different types of nodes,  $P$  nodes whose children's order can be permuted arbitrarily,  $Q$  nodes whose children can be reflected only, and leaves.

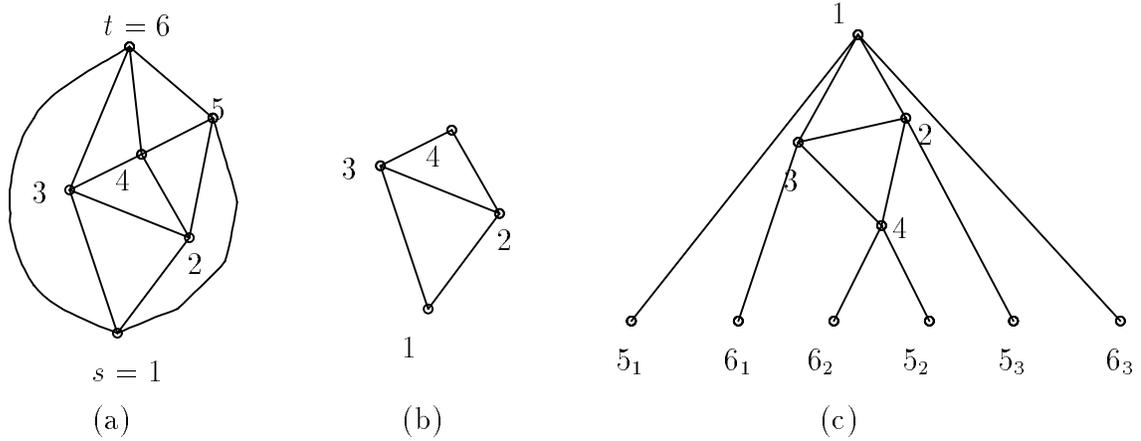


Figure 3.1:  $st$ -numberings of  $G$  (a) and  $G_4$  (b); bush-form  $B_4$  for  $G_4$  (c).

The main idea is to build a PQ-tree for the embedding:  $P$  nodes represent cut-vertices of the embedding thus far (components of cut-nodes can be permuted in any order around the cut-vertex);  $Q$  nodes represent 3-connected components (which can only be reflected around their separation pairs), and leaves represent virtual vertices. A bush form is an embedding of such a graph with all the virtual vertices on the outer face, as shown in Figure 3.1(a). Every planar graph  $G$  has a bush form for each  $G_k$ ,  $1 \leq k \leq n$  [4]. The algorithm of Booth and Lueker [4] (in addition to defining PQ-trees) determines the permutations and reflections

required to get all like-labelled vertices adjacent at the leaves of the structure so that they can be combined into one node without crossing nodes in the embedding. The embedding/testing thus consists of successively constructing bush forms  $B_1, B_2, \dots, B_n$ , and performing permutations on each to combine virtual vertices into single “real” vertices. An example of a bush form and its PQ-tree is given in Figure 3.2. Note that P-nodes are denoted by circles, Q-nodes by rectangles, and leaf nodes (virtual vertices) are unmarked.

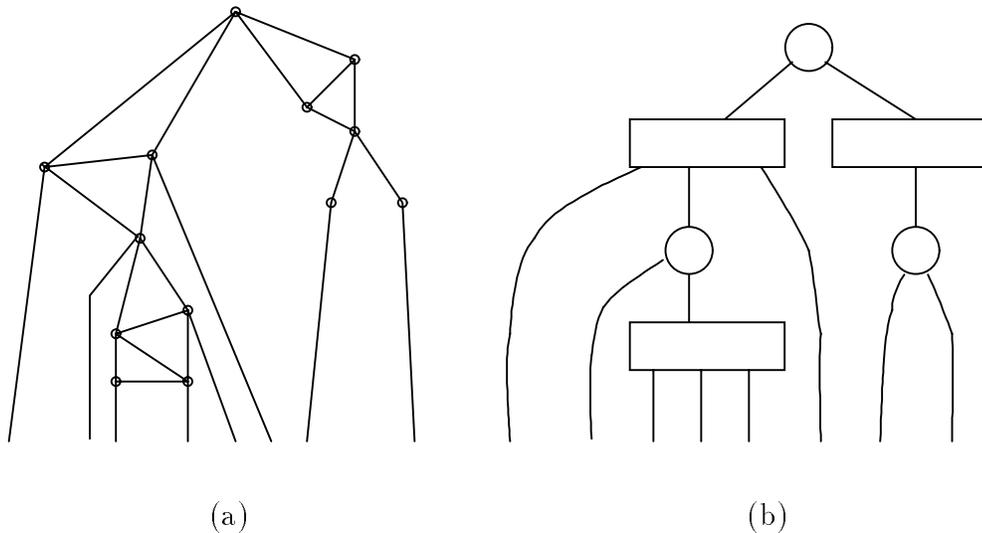


Figure 3.2: Bush form (a) and its associated PQ-tree (b).

## 3.2 Drawing Planar Graphs

A problem with the embeddings given by both the Hopcroft-Tarjan and the Lempel-Even-Cederbaum algorithms is that they can use possibly exponential area (grid-size) in the resulting drawing. Many vertices could be recursively embedded within successive faces of the output representation and a naive drawing algorithm would give impractical output. For general planar graphs, this is not optimal, however we

shall see in Section 3.4 that for upward planar graphs, we will (provably) have this problem with any drawing algorithm.

DeFrayssieux, Pach and Pollack [8] have described an  $O(n \log n)$  algorithm which takes an upward planar embedding (such as that from either planarity testing algorithm described) as input, and constructs a straight-line physical plane drawing on the  $2n - 4$  by  $n - 2$  grid (i.e. vertices are mapped to integer coordinates on the  $0 \leq x \leq 2n - 4$ ,  $0 \leq y \leq n - 2$ ,  $\mathbb{R}^2$  subplane, and edges are mapped to straight lines between corresponding vertices, such that no two edges intersect except at their endpoints). This upper bound on the area of plane graph drawings applies to all planar graphs. Chrobak [7] (referenced by Tamassia and Eades' bibliography [29]) has improved their algorithm to linear time. Independently, the same result (upper bound and linear time) was achieved by Schnyder [27], with the added feature that the co-ordinates are not "fixed", but have a "purely combinatorial meaning". DeFrayssieux, Pach and Pollack's paper also gives a (slightly) super-linear lower bound on the size of a grid required to support all planar graphs.

### 3.3 Testing for Upward Planarity

The problem of testing for upward planarity is not as well known, or as developed, as that of general planarity. Obviously, a given digraph can only be upward planar if it contains no directed cycles. DiBattista and Tamassia have proven that any upward planar graph is a subgraph of an (upward) planar  $st$ -graph [10]. An  $st$ -graph is a graph labeled with an  $st$ -numbering (defined in the previous section) consistent with its underlying partial order; i.e.  $u < v$  if and only if  $st\text{-number}(u) < st\text{-number}(v)$ . Platt [24] has shown that a single-source single-sink directed graph is upward planar if and only if the graph generated by adding a single edge from the source to the

sink is planar; this also follows from the Lempel-Even-Cederbaum theorem of the previous section. Thus it can be tested in linear time whether an  $st$ -graph has an upward plane drawing. However, the problem of determining whether a given digraph is a subgraph of a  $st$ -graph is open (ie. not known to be polynomial or NP-hard) [29], so the characterization is not a strong one.

The current work considers the case of only one source, but any number of sinks. If all sinks but one could be removed, then the problem would degenerate into the known case—this is exactly what our algorithm achieves. We find the problem to be more difficult than just modifying the Even-Tarjan algorithm, because more freedom is available for movement in the drawing when multiple sinks are allowed, and depth first search is not directly applicable in the directed case. Thomassen’s theorem (1.1) is our starting point. A reasonable amount of work will have to be done before it can be applied.

A similar result to ours is to test planarity of a “hierarchical” graph, in which the vertices of  $G$  are partitioned into  $k$  (a constant, usually given as part of the input) subsets, with no two vertices in any subset being connected by an edge in  $G$  (as in a subroutine call chart where all subroutines are assigned a “level”, and any subroutine can call others at only a strictly lower level), and the drawing must put all vertices at a lower level below all vertices of a higher level. A linear algorithm for this problem is given by DiBattista and Nardelli [11].

A very recent result of DiBattista, Liu and Rival is that any bipartite graph is upward-planar if and only if it is planar [9]. A graph  $G = (V, E)$  is *bipartite* if  $V$  can be partitioned into two disjoint subsets  $V_1$  and  $V_2$  such that every edge  $(u, v)$  of  $E$  connects  $u$  in  $V_1$  ( $V_2$ ) to  $v$  in  $(V_2)$  ( $V_1$ ) (i.e. no two vertices of  $V_1$  are joined by an edge, and no two vertices of  $V_2$  are joined by an edge).

### 3.4 Drawing Upward Planar Graphs

All known algorithms for drawing upward planar graphs make assumptions about the form of the graph. DiBattista and Tomassia give an  $O(n \log n)$  algorithm for determining a straight line drawing of a single-source single-sink graph; i.e. an  $st$ -graph. A monotonic grid drawing, where all edges are represented by polygonal lines with positively growing  $y$ -coordinate, can be constructed in linear time. The straight line drawing can have exponential grid-size, but the polygonal grid drawing guarantees no more than  $O(n)$  bends in the polygonal edges and a linear size grid.

The same result is achieved by Chiba and Nishizeki [6, 23] as a linear extension of Booth and Lueker's PQ-tree algorithm. The algorithm assumes an  $st$ -numbering which also realizes the partial order implied by the digraph  $G$ , and modifies the PQ-tree algorithm to retain direction while embedding the graph.

No polynomial algorithms are known for embedding a general directed upward-planar graph, or for determining if such an embedding exists [29].

A hierarchical graph (mentioned briefly in the previous section) can be drawn in linear time (Eades, Lin and Tamassia) [13].

An interesting result, also due to DiBattista, Tomassia, and Tollis [12] is the following, which shows that upward plane graphs do not have the same lower bounds on drawing area as undirected plane graphs:

**Theorem 3.4** [DiBattista, Tomassia, and Tollis] *There exists a class of planar digraphs that require exponential area in any straight-line upward grid drawing.*

We can claim the same result for single source DAGs, using the above, and the fact that every upward planar graph is a subgraph of a planar  $st$ -graph (the proof

of which [10] adds no more than  $O(n)$  vertices or edges in its construction). The same paper treats the issue of displaying symmetries and isomorphic subgraphs in upward planar drawings.

### 3.5 Graphs and Partially Ordered Sets

Platt's Theorem [24] states that any single-source single-sink digraph admits an upward planar drawing if and only if the underlying undirected graph with an edge added from the source to the sink is planar [24].

A *lattice* is a partially ordered set in which every two elements have a unique least upper bound or *join*, and greatest lower bound or *meet*. It is known that any planar poset with a least and greatest element is a lattice [21]. The theory of lattices is more developed than that of arbitrary posets (see [2]), which may be cause for algorithms for upward planarity testing of lattices to be “easier” than the more general case.

Although any directed graph corresponds to some partially ordered set  $P$ , the minimal graph which captures the ordering of  $P$  is called the *Hasse diagram* or *order graph* of  $P$ . A Hasse diagram has no transitive edges; ie.  $(u, v) \notin G$  if  $\exists w|(u, w)$  and  $(w, v) \in G$ .

We note that the problem of testing a Hasse Diagram for planarity is identical to that of testing an arbitrary digraph, since a Hasse Diagram is a digraph and, conversely, subdividing each edge of a digraph with a new vertex yields a graph with no transitive edges. Equivalence does not however hold for a drawing algorithm, since an upward plane straight-line drawing of a subdivided Hasse diagram would generate a monotonic grid drawing of the Hasse diagram itself (not

necessarily a straight-line drawing); however, this does immediately imply that the lattice-drawing algorithm of DiBattista and Tamassia [10] can find a monotonic grid drawing of an arbitrary  $st$ -graph in linear time, such that the resulting drawing will have no more than  $n$  bends [10].

For a given partial order  $P$ , one can always determine a *linear extension*  $Q$  of  $P$ , i.e. any total order which includes (is consistent with)  $P$ . There always exists such a linear extension, and the intersection of all linear extensions of  $P$  is  $P$  itself [32]. A minimal set  $R$  of linear extensions of  $P$  whose intersection is  $P$  is called a *realizer* of  $P$  and the *dimension* of  $P$  is the minimum cardinality of any realizer of  $P$ .

Various work has been done on determining a characterization of upward planarity through poset theory rather than graph theory; a survey of the work has been done by Trotter [32]. Some of the more interesting results are the following:

Kelly and Rival [21] have shown that there exist planar posets of arbitrary dimension. This is significant, because it was originally thought that the dimension of a poset could be related to the minimum genus of a surface upon which its Hasse diagram could be embedded. The same line of research led Trotter [32] to determine that embedding Hasse diagrams on the sphere is “different” from embedding them in the plane. These results show that upward planarity has a more complicated structure than planarity.

Work on specific issues in the drawing of lattices, such as minimizing the set of edge slopes required for a plane embedding, is discussed in Tamassia and Eades’ Bibliography [29].

A *spherical order* is an order with a unique top and bottom element whose Hasse diagram can be embedded on the sphere with no crossing arcs. This class differs

from the class of planar orders, as mentioned above and in the requirement for a least and greatest element. Note that every planar lattice is a spherical order (Kelly and Rival) [21]. A *truncated spherical order* is a spherical order with its bottom and top elements removed. Foldes, Rival and Urrutia [17] show that an ordered set is spherical if and only if it has a bottom and a top element and its Hasse diagram is planar; that an ordered set is truncated spherical if and only if it has a light source representation, and that there is a linear algorithm to determine if an ordered set has a light source representation. A light source representation is a “blocking” relation generated by a point light source, and a set of objects embedded in the plane. These properties of “light source order” and “sphericity” are hence said to be *invariant* for lattice diagrams. Rival [26] discusses possible invariant properties for order diagrams (no non-trivial invariant is known).

# Chapter 4

## Strongly-Equivalent Upward Planarity

We wish to consider the following sub-question: Given an acyclic digraph  $G$  with a single source  $s$  and a planar representation  $\overline{G}$  of  $G$ , does  $G$  admit an upward planar representation  $\Gamma$  strongly equivalent to the given embedding  $\overline{G}$ ? That is, can we contort  $\overline{G}$  slightly to make it a representation of an upward plane drawing without changing any of the facial cycles or the outer face?

Thomassen’s Theorem (1.1) provides the requirements for checking this, but no efficient algorithm immediately follows. Here we provide an alternative statement of Thomassen’s result in the form of an optimal (linear) algorithm to test  $G$  for upward planarity strongly equivalent to some given embedding  $\overline{G}$ .

Recall that an upward planar embedding is, in effect, a planar embedding with additional dummy edges which “resolve” those sinks not on the outer face—which connect them to another vertex, so that the graph is still upward planar, but the sink vertex is no longer a sink. We claim that given such a representation, it is

easy to draw an upward planar graph. How to do so will be addressed in Chapter 8. The algorithm of this chapter will output an upward planar embedding of  $G$  strongly equivalent to  $\overline{G}$ , if one exists, by adding edges to  $\overline{G}$ .

We define a *violating cycle* of  $G$  as a cycle  $C$  which has no vertex  $v$  which is dominated by both its neighbours in  $C$  and which is not the tail of any inward edge from a vertex on  $C$  with respect to the embedding  $\overline{G}$ . The algorithm finds either a strongly equivalent upward plane representation or a violating cycle and it is clear that we cannot have both so the proof of correctness for algorithm also provides a proof of the theorem.

The idea of the algorithm is the following: we find a sink  $v$  on the outer face of  $\overline{G}$ , delete it, recursively find an upward planar embedding of  $G \setminus v$  strongly equivalent (with respect to the original edges of  $G$  only) to  $\overline{G} \setminus v$ , and then return  $v$  to the outer face of this upward planar embedding.

The input is a single-source digraph  $G$  and a planar embedding  $\overline{G}$  of  $G$  with the source on the outer face. The algorithm returns either an upward planar embedding  $\Gamma$  of  $G$  such that  $\Gamma$  is strongly equivalent (with respect to the original edges only) to  $\overline{G}$  or a violating cycle  $C$ .

Define an *outer-facial walk* of a connected planar graph  $G$ , with respect to some planar representation  $\overline{G}$ , as a cyclic walk induced by the outer face of  $\overline{G}$ . The outer facial walk  $W$  of a single source DAG  $G$  with its source  $s$  on the outer face is assumed to travel from  $s$  to  $s$ , so, given  $\overline{G}$  (which specifies the cyclic order of vertices around  $s$ ), the outer facial walk we are concerned with is unique. Within the sequence of vertices and edges which specify the outer facial walk, there exist one or more cycles—consisting of a subsequence of the walk starting at one vertex, called the *attachment* of the cycle, and returning to the same vertex, with no other

vertices or edges repeated. In the simple case, the outer facial walk is itself a cycle from  $s$  to  $s$ .

**Algorithm:** If there is no sink on the outer facial walk  $W$  of  $\overline{G}$ , then return a violating cycle  $C$  as follows: If  $W$  is itself a cycle, then it is the (only) violating cycle, as it contains no sink (and hence no vertex can satisfy the conditions for “ $v$ ”). If  $W$  is not a cycle, consider the cycles existing within  $W$  as discussed above. If any one of these cycles  $C$  within  $W$  contains  $s$  as its attachment to  $W$ , then it is violating, as all vertices on  $C$ , other than  $s$ , are non-sinks, and  $s$  is not a sink. Otherwise, choose some cycle  $C$  within  $W$  which does not contain  $s$ , and let  $u$  be its attachment to  $W$ . Then  $u$  is a source in  $C$ ; if not, the component  $H$  of  $G \setminus u$  with  $C$  as its outer face would have to contain a source other than  $u$ , and  $s$  is not in  $H$ , so  $G$  has more than one source—a contradiction. This same issue is treated more rigorously in the proof of Lemma 5.1 on page 28.

Otherwise (there is a sink on  $W$ ), let  $v$  be the first sink on  $W$ . Recursively test whether  $G \setminus v$  has an upward planar embedding strongly equivalent to  $\overline{G} \setminus v$ . If not, return the violating cycle of  $\overline{G} \setminus v$  which is also a violating cycle contained in  $\overline{G}$ . If an upward planar embedding  $\Gamma_v$  is returned then add  $v$  into the outer face of  $\Gamma_v$  to form  $\Gamma$  by connecting it with a directed edge from all adjacent (in  $G$ ) vertices and all sinks on  $W$  between the first and last such adjacent vertex (this is the resolving of the internal (no longer on the outer face) sinks mentioned in Chapter 2). The latter edges (from sinks) are marked as “dummy” edges as they are not part of the original  $G$ . Since  $\overline{G}$  has  $v$  on its outer face, and  $\overline{G} \setminus v$  has a strongly equivalent upward plane embedding  $\Gamma_v$ , all vertices of attachment of  $v$  and all these sinks are on the outer face of  $\Gamma_v$ , so  $v$ , and all edges between  $v$  and  $G \setminus v$  can be added to the outer face of  $\Gamma_v$  without destroying planarity. Since all internal sinks have been “resolved” with dummy edges, the result  $\Gamma$  is an upward planar embedding

strongly equivalent to  $G$ .

Note that when we are referring to equivalence and strong equivalence of embeddings the extra dummy edges are not counted; they obviously change the faces of the graph, but do not affect the properties we are interested in (which are determined by the (unchanged) outer face, once an upward planar embedding is found).

Since  $s \xrightarrow{+} v$  for all vertices in  $V(G)$  (i.e.  $s$  is the single source of  $G$ ), and  $s$  is never deleted while there is any other vertex on  $W$ , the subproblems retain the single source property. Since  $v$  can never have an edge pointing downward in the cycle,  $G$  remains acyclic, even with the extra edges.

To prove that the algorithm is correct it remains to note that if at some stage, say for the induced graph  $G - S$ , no sink is found on the outer face of  $\overline{G} - S$  then since none of the vertices inside this outer face have yet been deleted the outer face forms a violating walk, and contains a violating cycle as discussed above.

Since the operations necessary to implement this algorithm in linear time may not be obvious, the following section gives a more detailed discussion of the data structures and operations involved and a proof of the time bound.

## 4.1 Implementation Details

Since we can use the Hopcroft Tarjan algorithm to compute a planar embedding in linear time, we will assume that the input is a collection of nodes, with vertices adjacent to a given node in circular (counter-clockwise) order that occur in the embedding around that node, and a specification of the outer face.

We implement  $W$  (of the previous section) as a doubly linked list, chained through the graph  $\overline{G}$ , which is a collection of nodes and (doubly linked) pointers

to adjacent nodes. We assume that the input comes in this form, with the list of adjacent vertices to a given node in circular order around that node's embedding in the plane. Define an additional list  $D$ , called the **candidate list**, which contains a list of all sinks in  $W$ . This list is also doubly linked, and shares nodes with  $\overline{G}$  and  $W$ . We thus have a single copy of each node, and a linear number of pointers in total, so the space is  $O(n)$ . Assume that the incoming and outgoing degree of each node is stored with the node, and updated as edges are deleted, so that a node can be determined to be a sink or source in constant time.

**Algorithm S-E-UP-PLANAR:**

**I.** Choose the first vertex  $v$  from  $D$ .

(i) If  $D$  is empty,  $G$  is not upward planar, so return a violating cycle within  $W$  as described in the previous section. If  $v = s$ , then  $G$  is trivially upward planar, so just return an embedding of the single node.

(ii) If  $\deg^-v = 1$  (edge  $(u, v)$ ), then simply delete  $v$  from  $G$ ,  $W$ , and  $D$ , and also delete the extra copy of  $u$  in  $W$ . If  $u$  is a sink in  $G \setminus v$ , then add  $u$  to  $D$  as well, yielding  $G'$ ,  $W'$  and  $D'$ .

(iii) Otherwise  $\deg^-v > 1$ . For each edge  $(u, v)$ , starting with the one preceding  $v$  in  $W$ , add all new vertices in the face containing  $v$  to  $W$  and any sinks that are discovered to  $D$ . Delete  $v$  from  $G$ ,  $W$ , and  $D$ , yielding  $G'$ ,  $W'$  and  $D'$ .

**II.** Recursively determine an upward plane embedding or violating cycle of  $G \setminus v$ , using  $G'$ ,  $W'$  and  $D'$ .

**III.** Replace  $v$  in cyclic order around its vertices of attachment in  $\overline{G'}$ , giving  $\overline{G}$ . The order does not change from the original embedding. At the same time, traverse the vertices of  $W$  covered by  $v$ —those which are on  $W'$  but not on  $W$ , and add an edge from any sink covered by  $v$ , as mentioned in the previous section.

Each node in  $G$  can be added to  $W$  at most once, by planarity. Since adjacent nodes are sorted in circular order, no searching need be done, so the total time is linearly proportional to the number of nodes in  $G$ , namely  $O(n)$ . The two interesting cases ((ii) and (iii) above) are illustrated in Figure 4.1(a):  $v_1$  is case (ii), and  $v_2$  is (iii). Figure 4.1(b) shows the addition step on  $v$  after the sub-embedding is returned.

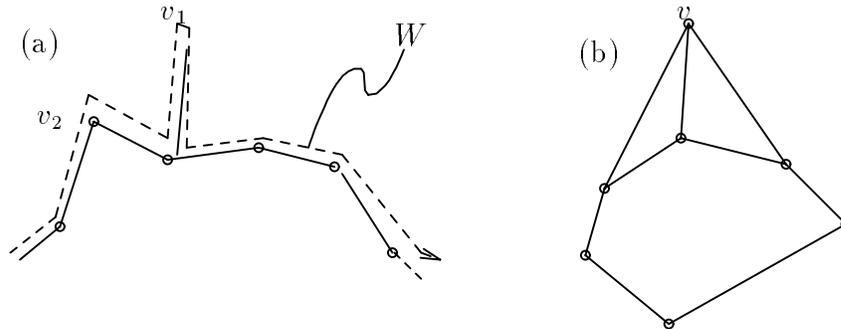


Figure 4.1: Cases for  $v$  in Algorithm S-E-UP-PLANAR.

The standard abstract planar embedding of a graph is to have each node in  $G$  contain a list of the vertices appearing in circular order around it. Our algorithm determines if such an embedding extends to an upward embedding with the same facial structure, but with (possibly) some number of new edges from internal sinks to vertices which first cover them. Thus, the output from this algorithm is either failure (and a violating cycle) or the modified abstract upward embedding.

This result is exactly that of Theorem 1.1, but the proof is algorithmic, and the algorithm is provably linear time.

# Chapter 5

## Separation into Bi-Connected Components

**Lemma 5.1** *A DAG  $G$  with a single source vertex  $s$  and a cut vertex  $v$  admits an upward plane drawing iff each of the  $k$  components  $H_i$  of  $G$  (with respect to  $v$ ) admits an upward plane drawing.*

**Proof:** The necessity is obvious; any subgraph of  $G$  must have an upward plane drawing if  $G$  does. For the converse, we have 2 cases,  $s = v$  and  $s \neq v$ .

Suppose  $s = v$ , and all  $H_i$  have upward plane drawings  $\overline{H_i}$ . Since  $v$  is the single source of  $G$ , it is also the single source of each  $H_i$ , and is the lowest point on the outer face in any upward plane drawing. We complete the embedding of  $G$  by dividing the half-plane above  $v$  into  $k$  “pie shaped” segments, each containing one component, with  $v$  identified at the same point in  $\overline{G}$ . The construction is illustrated in Figure 5.1(a).

If  $s \neq v$ , then identify the component of  $G$  with respect to  $v$  containing  $s$  as  $H_1$ . All other components ( $i > 2$ ) must have  $v$  as a single source, otherwise  $G$

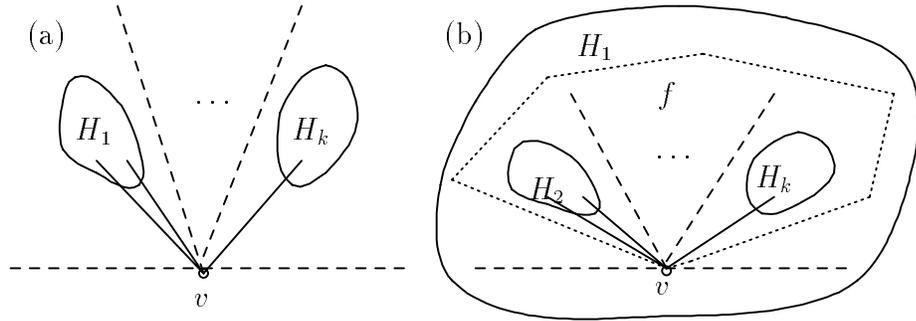


Figure 5.1: Construction of  $\overline{G}$  from its bi-connected components

has more than one source. If some component other than  $H_1$  had a vertex  $x < v$ , then either that component contains a second source in  $G$  or  $G$  contains a cycle as  $s \xrightarrow{+} v \xrightarrow{+} x \xrightarrow{+} v$ . Suppose there is an upward plane drawing  $\overline{H_1}$  of  $H_1$ . Consider some face  $f$  of  $\overline{H_1}$  containing some portion of the half-plane above  $v$ . Then all other components can be embedded into the  $k$  segments of this face as they were added to the open face in the first case. This is illustrated in Figure 5.1(b). As before, any upward drawing must have its single source as the lowest point on the outer face. ■

Note that the operation of inserting the components  $H_i$  within faces of  $H_1$ , or into the regions of the plane around the half-plane above  $v$  in an embedding of  $H_1$ , will require scaling of the inserted sub-embeddings. Any scaling will preserve both upward planarity and the outer face of the “inserted” graphs, but the problem of “bunched up” vertices or precise real arithmetic is not addressed.

Tarjan’s well known algorithm [28] finds the cut-vertices and biconnected components of a graph in linear time, and can be used as a subroutine for our algorithm.

## Chapter 6

# Separation into Tri-Connected Components

The algorithm of Chapter 3 tests for upward planarity of  $G$  by examining a strongly-equivalent planar embedding  $\overline{G}$ . In principle, we could apply this test to all planar embeddings of  $G$ , but this would take exponential time. In order to maintain a tractable number of embeddings, we will first isolate the triconnected components of the input graph, each of which has a unique planar embedding under equivalence, and only a linear number of planar embeddings under strong equivalence [3]. Since we will perform the splitting and merging of triconnected components in quadratic time, the total time will then be quadratic.

This step is similar in spirit to the previous operation of splitting biconnected components and merging the recursively calculated drawings. However, the degree of complication is much greater, because the cut-set vertices impose restrictive structure on the merged graph. In the previous section describing splitting into biconnected components, it was sufficient to simply test each component separately,

since biconnected components did not interact in the combined drawing; this is not the case for triconnected components, as illustrated by the two examples in Figure 6.1. In (a), the union of the graphs is upward planar, but adding the edge  $(u, v)$  to each makes the second component non-upward-planar. In (b), the graph is non-upward-planar, but each of the components is upward planar with  $(u, v)$  added. The problem of specifying necessary and sufficient conditions for the split and merge is nontrivial. Recall our convention that direction arrow-heads are assumed to be “upward” unless otherwise specified.

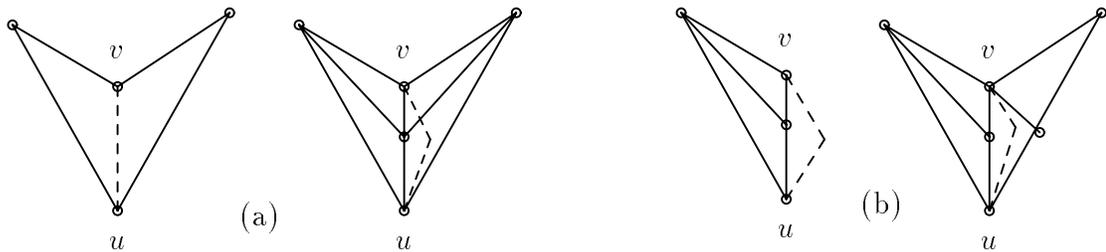


Figure 6.1: Added complication of 2-vertex cut-sets.

We will find it convenient, particularly for the case where the source  $s$  is in a separation pair, to split the graph into exactly two pieces at separation pairs.

There are two main issues. Firstly, we must identify which component will be the “outer” component, because this imposes restrictions on the other (“inner” component) to adapt to its facial structure (in order to be injected within a face). It will always be true that the inner component will have more restrictions upon its embedding, because it must fit within the prescribed face. Specifically, a list of vertices will be required to be on the outer face of any embedding to retain planarity in the merge. Secondly, we must be able to properly represent the facial structures of the two components to ensure that the recursively calculated embeddings can be merged without destroying upward planarity.

Our general subproblem instance consists of a biconnected graph  $G$ , and a set

of vertices  $X = \{x_i\} \subseteq V(G)$  which will be required to be on the outer face of any planar embedding of  $G$ .  $G$  is broken up into two components at a cut-set  $\{u, v\}$ , and recursive calls made. We will give conditions based on the type of cut-set involved as to whether upward plane drawings of the two components can be put back together into an upward plane drawing of the whole. These conditions are broken into three cases: where  $u$  and  $v$  are incomparable,  $u$  and  $v$  are comparable with  $u \neq s$  (i.e.  $s < u < v$ ), and where  $u, v$  are comparable with  $u = s$ .

The conditions prescribed will be in the form of *markers* added to each component to represent the shape of the other component in the decomposition. If the graph were undirected, it would be sufficient to add a single edge between the cut vertices in each component, because the only requirement would be that the vertices share a face. We would also not need to require any vertices to be on the outer face, because any face can be made the outer face. This is not true for upward planarity. The type of markers needed will depend on the particular graph.

## 6.1 Some Basic Results

The following are some preliminary results and definitions useful throughout this section. Recall our convention that *connected*, *biconnected* and *triconnected* are defined in terms of the underlying undirected graph, and that  $\cup$  refers to the the *inclusive* union of graphs.

### 6.1.1 Facial Structure of Upward Planar Graphs

We would first of all like to get an intuitive idea of what the faces of upward plane drawings must really look like; See Figure 6.2. The outer face of any upward plane

diagram must have a “similar” shape to the upward facing hand shown in Figure 6.2(a), because there is only one source  $s$ , which must be at the bottom of the outer face and there can be any number of sinks. An inner face must have a shape “similar” to the downward facing hand in Figure 6.2(b): if there were more than one vertex like  $w$  on the upper side of the face, then there would have to be an extra source between them. Note that both cases degenerate to the shape of Figure 6.2(c).

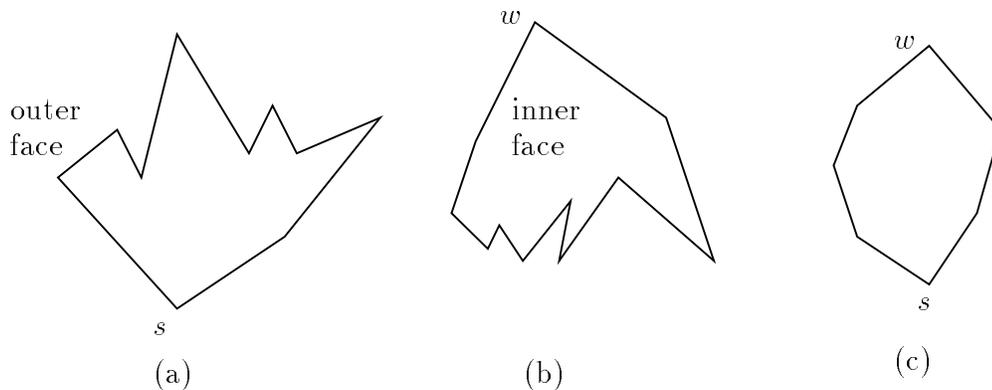


Figure 6.2: Facial structure in an upward plane drawing.

**Definition 6.1** Let  $G$  be a biconnected DAG with a single source  $s$  and separation pair  $\{u, v\}$ , where  $u, v \neq s$ . Then, the connected component of  $G$  with respect to  $\{u, v\}$  containing  $s$  is called the **source component** of  $G$  with respect to  $\{u, v\}$ .

### 6.1.2 Properties of Cut Sets in Digraphs

We can now make the following two observations.

**Proposition 6.2** If  $G$  is a single source connected DAG with a cut-set  $\{u, v\}$ , where  $u$  and  $v$  are incomparable, then  $u$  and  $v$  have in-degree 0 (i.e. are sources) in all components of  $G$  with respect to  $\{u, v\}$  except the source component.

**Proof:** Note that  $G$  is connected, otherwise it would have more than one source. Also, if any vertex  $x$  is incomparable with the source, then  $G$  would have more than one source. This implies that neither  $u$  nor  $v$  is a source in the source component.

Consider any component  $H$  of  $G$  with respect to  $\{u, v\}$ , not containing the source  $s$ . In  $G$ , any vertex  $x$  of  $H$  is connected by a directed path from  $s$  (only) through either  $u$  or  $v$ , as  $\{u, v\}$  separates  $s$  from  $H$ . Thus,  $u < x$  or  $v < x$ , for all vertices  $x \in H$ . Without loss of generality, suppose that  $v$  has an incoming edge  $(x, v)$ ; then, either  $u < v$  or  $H$  (and hence  $G$ ) contains a cycle  $v \xrightarrow{+} x \rightarrow v$ , which are both contradictions. ■

**Proposition 6.3** *If  $G$  is a connected DAG with exactly two sources  $u$  and  $v$ , then there exists some  $w_t$  such that two vertex disjoint (except at  $w_t$ ) directed paths  $u \xrightarrow{+} w_t$  and  $v \xrightarrow{+} w_t$  exist in  $G$ .*

**Proof:** Let  $G$  be such a DAG and let  $P$  be an undirected path from  $u$  to  $v$ . Note that every  $x$  in  $P$  is comparable with either  $u$  or  $v$ , otherwise  $G$  has more than two sources. Follow  $P$  from  $u$  to the first node  $x$  (following  $y$  on  $P$ ) incomparable with  $u$  (in  $G$ ). Then,  $x$  is comparable with  $v$  and  $(x, y)$  is an edge in  $G$  (otherwise  $u < x$ ), so  $y$  is also comparable with  $v$ . Taking the first common vertex in the paths  $u \xrightarrow{+} y$  and  $v \xrightarrow{+} y$  gives  $w_t$ . ■

We are now in the position to discuss the structure of components arising from 2-vertex cuts. Specifically, the following results show the existence of lower bounds and upper bounds (in the partial order corresponding to  $G$ ) under certain conditions. This allows us to prove the necessity conditions of the next section, because the existence of such structures indicate that certain subgraphs must exist as subgraphs within  $G$ . It is these subgraph properties which will be used to show that the component-markers are appropriately chosen.

**Lemma 6.4** *If  $G$  is a biconnected DAG with a single source  $s$ , and  $u$  and  $v$  are incomparable vertices in  $G$ , then there exists some  $w_s$  such that two vertex disjoint (except at  $w_s$ ) directed paths  $w_s \xrightarrow{+} u$  and  $w_s \xrightarrow{+} v$  exist in  $G$ . If  $\{u, v\}$  is a cut-set in  $G$ , then there also exists some  $w_t$  such that two vertex disjoint (except at  $w_t$ ) directed paths  $u \xrightarrow{+} w_t$  and  $v \xrightarrow{+} w_t$  exist in  $G$ .*

**Proof:** Since  $G$  is a single source digraph, there exist directed paths from  $s$  to  $u$  and  $s$  to  $v$  in  $G$ , so if these paths are vertex disjoint then  $w_s = s$  and we are done. Otherwise, there is some maximal element (in the orders imposed by the two chains on the elements within them) which is shared by both paths, and this suffices for  $w_s$ .

For the existence of  $w_t$ , let  $u$  and  $v$  be an incomparable separation pair of  $G$ . Since  $\{u, v\}$  cuts  $G$  into at least two connected components, any non-source component  $H$  has  $u$  and  $v$  as its (exactly) two sources, and the result follows from Proposition 6.3. ■

**Lemma 6.5** *If  $G$  is a biconnected DAG with a single source  $s$  and cut-set  $\{u, v\}$ , where  $u < v$  in  $G$ , then in any non-source component  $H$  of  $G$  with respect to  $\{u, v\}$ , where  $\deg^+ v > 0$ , there exists some  $w_t$  such that  $u \xrightarrow{+} w_t$  and  $v \xrightarrow{+} w_t$  are vertex disjoint directed paths in  $H$ .*

**Proof:** No vertex other than  $u$  and  $v$  can be a source in  $H$ , otherwise  $G$  has more than one source.  $u$  is always a source in  $H$ , otherwise  $G$  contains a directed cycle. If  $v$  is also a source, then we are done by Proposition 6.3.

If  $v$  is not a source, let  $w \in H$  be a vertex dominated by  $v$ .  $G$  is biconnected, so there are two vertex disjoint  $u \xrightarrow{+} w$  undirected paths in  $G$ . But  $u$  and  $v$  are cut-vertices in  $G$ , so at least one of the paths  $P$  lies completely within  $H$  and does

not contain  $v$  (as  $w$  is in  $H$  and the only exit points from  $H$  are  $u$  and  $v$ ). Every  $x$  on  $P$  is comparable with either  $u$  or  $v$ , or else  $G$  has more than one source. Find the last vertex  $y$  on  $P$  which has a  $u \xrightarrow{+} y$  path (in  $G$ ) without  $v$ . If  $y = w$ , then we are done. Otherwise, the vertex  $x$  following  $y$  on  $P$  has any  $u \xrightarrow{+} x$  path necessarily going through  $v$ . Then there exist directed paths  $v \xrightarrow{+} x$ ,  $u \xrightarrow{+} x$  with the latter not containing  $v$  so the last common vertex on these paths provides a  $w_t$ . ■

### 6.1.3 Topological Properties of Upward Planar Graphs

We now prove that certain contractions and transformations on a digraph preserve the property of upward planarity. The first result is used to simplify the structure of graphs in the decomposition/merging phases; the remaining results are necessary for proofs in the next section.

**Definition 6.6** *A subdivision vertex of  $G$  is any vertex  $v$  which satisfies  $\deg^+v = \deg^-v = 1$ . A directed subdivision is a subdivision of  $G$  which preserves edge-transitivity (i.e. inserting  $w$  into  $(u, v)$  can only yield  $(u, w), (w, v)$ , so the property  $u < v$  is maintained).*

Note that subdivision vertices add no information or structure to the problem of testing for upward planarity as proven by Thomassen [30].

**Lemma 6.7**  *$G$  admits an upward plane drawing if and only if any directed subdivision of  $G$  admits an upward plane drawing.*

As a result, we can preprocess  $G$  to contract long paths of a directed subdivision into single edges in the following way: For any path  $P = v_0v_1 \dots v_k$  of  $G$  with  $k \geq 2$

and  $\deg^+v_i = \deg^-v_i = 1$  ( $i = 1 \cdots k - 1$ ), replace  $P$  by the single edge  $v_0v_k$ . The identification of such nodes can be done with a single depth-first search of the graph, collapsing edges as the search proceeds, and thus takes linear time.

Using this contraction process before process of isolating triconnected components will allow us to assume that any non-source, non-sink nodes have degree at least three (or they would have been contracted).

Another result is the following, which also implies Lemma 6.7:

**Lemma 6.8** *Let  $G$  be a DAG and  $v$ , dominated by  $u$ , be a vertex of  $G$  with in-degree 1. Then,  $G/(u, v)$  is upward planar if  $G$  is upward planar.*

**Proof:** Suppose  $\overline{G}$  is some upward planar embedding of  $G$ . Then, there is an  $\varepsilon$ -region or “corridor” around the edge  $(u, v)$  down which  $v$  and its monotonic upward-directed edges can be contracted without destroying planarity. The partial order through  $v$  is preserved since  $v$  has no other incoming edges, so there are no vertices between  $u$  and  $v$  in the partial order. See Figure 6.3(a). ■

**Corollary 6.9** *The same result holds for  $G$  and edge  $(u, v)$  with  $\deg^+u = 1$  by symmetry.* ■

We can now extend Lemma 6.7 to cover arbitrary digraphs rather than single vertices.

**Lemma 6.10** *Let  $G$  be a connected DAG with an edge  $(u, v)$  and  $H$  be a connected DAG with a single source  $u'$  and a sink  $v'$ . Let  $G'$  be  $G$  with the edge  $(u, v)$  subdivided by  $u'$  and  $v'$  (i.e.  $(G \setminus (u, v)) \cup (u, u') \cup (u', v') \cup (v', v)$ ), and containing  $H$  attached at  $u'$  and  $v'$ . Then,  $G'$  is upward planar if and only if  $G$  is upward planar and  $H$  is upward planar with  $u'$  and  $v'$  on the outer face.*

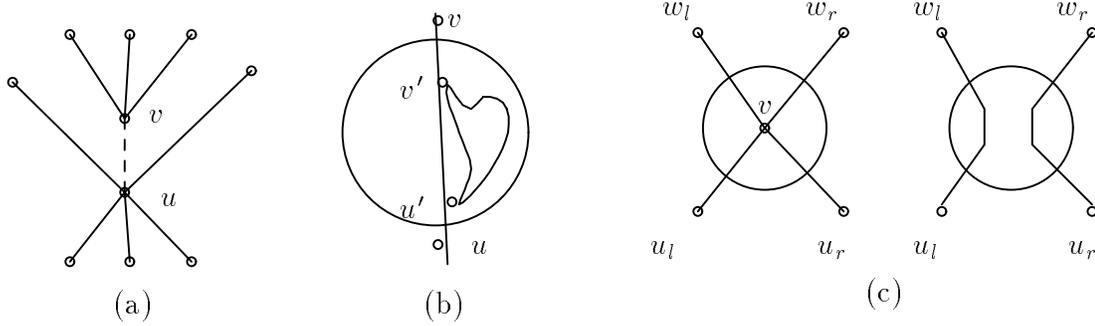


Figure 6.3: Transformations preserving upward planarity.

**Proof:** Since  $u'$  and  $v'$  are forced to be on the outer face of an upward planar embedding of  $H$ , and  $u'$  is the (only) source of  $H$ , and  $v'$  is a sink, an upward plane embedding  $\overline{H}$  can be inserted beside the edge  $(u, v)$  along an  $\varepsilon$ -corridor without destroying upward planarity. For the other direction,  $G$  is upward planar because it is homeomorphic to a subgraph of  $G'$ .  $H$  is upward planar as it is a subgraph of  $G'$  and the source  $u$  and all sinks share the outer face, so the edge  $(u, v)$  can be added without destroying planarity—the source can always “see” any sink on the outer face in an upward planar embedding. See Figure 6.3(b). ■

A similar result to subdividing an edge as just discussed, is to extend the digraph beyond (above) a sink node; where there is no edge (i.e. by adding an edge or graph with a connection to a the sink):

**Lemma 6.11** *Let  $G$  be an upward planar digraph containing a sink  $u$ . Then for some new vertex  $v$ ,  $G \cup (u, v)$  is also upward planar. Similarly for an upward planar digraph  $H$  with single source  $v$ ,  $G \cup H \cup (u, v)$  is upward planar.*

**Proof:** Since  $u$  is a sink in  $G$ , in any drawing  $\overline{G}$  of  $G$ , there is an epsilon ball in the half-plane above  $u$  containing no other vertices or edges. Inserting  $v$  into this area, and adding the edge  $(u, v)$  retains upward planarity. By repeated applications

of this operation of inserting edges, and and the previous topological results such as subdivisions, any single source upward planar graph can be inserted (“grown”) above  $u$  in  $G$ . ■

Note that this does not address the problem of arithmetic precision or scaling; only the issue of upward planarity.

**Lemma 6.12** *Let  $G$  be a DAG and  $v$  be a vertex of  $G$  with  $\deg^+v = \deg^-v = 2$ . Let  $\overline{G}$  be an upward plane embedding of  $G$  with  $u_l$  and  $u_r$  the left and right dominating vertices of  $v$  respectively, and  $w_l, w_r$  the left and right vertices dominated by  $v$  respectively. Then  $G \setminus v \cup \{(u_l, w_l), (u_r, w_r)\}$  is upward planar as well.*

**Proof:** For some sufficiently small  $\varepsilon$ , the  $\varepsilon$ -ball around  $v$  contains four regions, defined by the four edges incident upon  $v$ . The leftmost (rightmost) two can be joined into a single monotonic edge to the left (right) of  $v$ , and  $v$  deleted to give an upward plane embedding of  $G \setminus v \cup \{(u_l, w_l), (u_r, w_r)\}$ . See Figure 6.3(c). ■

## 6.2 Marker Components

In order to effectively decompose our input graph for testing purposes, we will find it necessary to reduce the biconnected components to their simplest relevant structure. An elementary step in this operation consists of splitting the input graph into exactly two components, replacing each component by a marker component in its companion, and recursively testing the two for upward planarity. The crucial point is to choose the appropriate marker, so as to sufficiently represent the structure of the companion. The markers are necessary for three reasons: Firstly, to ensure that the original graph is upward planar if and only if the two components (with

markers) are upward planar; secondly, to maintain biconnectedness; and thirdly, to maintain the single source property. The markers we will be interested in are shown in Figure 6.4. An important point to make at this time is that the markers, except

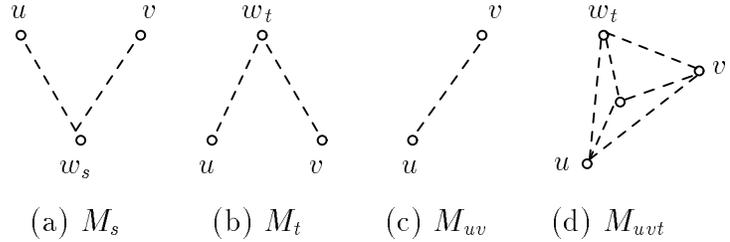


Figure 6.4: Marker Graphs.

for  $M_{uv}$ , are attached to the graph at only two vertices, which means that  $\{u, v\}$  still constitutes a cut-set. For the purposes of determining cut-sets and making recursive calls, each marker should be treated as a distinguished edge—a single edge labelled to indicate its role. As long as the type of marker is identified, the algorithm can continue to treat the vertices of attachment as source, sink or neither as appropriate for the particular operation.

The following three subsections describe the markers to use for different types of cut-sets  $\{u, v\}$ .

### 6.3 The cut-set $\{u, v\}$ , where $u$ and $v$ are incomparable

**Theorem 6.13** *Let  $G$  be a biconnected directed acyclic graph with a single source  $s$  and let  $X = \{x_i\} \subseteq V(G)$  be a set of vertices. Let  $\{u, v\}$  be a separation pair of  $G$ , with  $u$  and  $v$  incomparable, where  $S$  is the connected component of  $G$  with respect to  $\{u, v\}$  containing  $s$ , and  $H$  is the union of all other components. Then,*

$G$  admits an upward plane drawing with all vertices of  $X$  on the outer face if and only if

- (i)  $S' = S \cup M_t$  admits an upward plane drawing all any vertices of  $X$  in  $S$  on the outer face, and  $w_t$  on the outer face if some  $x \in X$  is contained in  $H$ .
- (ii)  $H' = H \cup M_s$  admits an upward plane drawing with all vertices of  $X$  in  $H$  on the outer face.

**Proof:** (Necessity) Suppose  $G$  admits an upward plane drawing with all  $x_i \in X$  on the outer face. By Lemma 6.4 there exist vertex disjoint paths  $P_u = u \xrightarrow{+} w_t$  and  $P_v = v \xrightarrow{+} w_t$  in  $H$ , for some  $w_t$ . If  $G$  is upward planar, then any subgraph of  $G$  is, so  $S \cup P_u \cup P_v$  is upward planar. But this is a directed subdivision of  $S'$ , so  $S'$  is also upward planar by Theorem 6.7. Similarly, there is some  $w_s$  in  $S$  such that  $H' = H \cup \{(w_s, u), (w_s, v)\}$  is homeomorphic to a subgraph of  $G$  and is upward planar.

Since all  $x_i \in X$  are on the outer face of  $\overline{G}$ , they are certainly on the outer face of the sub-embeddings  $\overline{S'}$  and  $\overline{H'}$  of the upward embedding  $\overline{G}$ . If some  $x_i \in H$  is on the outer face of  $\overline{G}$ , then the sub-embedding of the  $H$  portion of  $\overline{G}$  is within the outer face of the sub-embedding of  $S$ , so an embedding of  $S \cup M_t$  will have  $w_t$  on the outer face.

(Sufficiency) Suppose  $S'$  and  $H'$  admit upward plane drawings  $\overline{S'}$  and  $\overline{H'}$  with all  $x_i \in X$  on their respective outer faces and  $w_t$  on the outer face of  $\overline{S'}$  if some  $x_i \in X$  is contained in  $H$ . Since  $u$  and  $v$  have in-degree 0 in  $H$  (Lemma 6.2), and  $H$  has no other sources, all of  $H$  lies in some region above  $u$  and  $v$  in any plane embedding (since the added source  $w_s$  dominates both  $u$  and  $v$  and forces them to be on the outer face, and the lower portion of that face). Rename  $u$  and  $v$  in  $H$  as  $u'$  and  $v'$  respectively and consider  $S' \cup H' \cup (w_s, w_t)$ . Since  $\deg^+ w_t = 0$  (in  $S'$ ), this

combined graph is upward planar (Lemma 6.11). Applying Lemma 6.8 to contract  $(w_s, w_t)$  into a single vertex  $w$  and splitting  $w$  (which has in-degree and out-degree two) into two edges  $(u, u')$  and  $(v, v')$ , by Lemma 6.12, gives an upward planar graph  $S \cup H \cup \{(u, u'), (v, v')\}$ . But  $\deg^- u' = 1$  and  $\deg^- v' = 1$ , so by Lemma 6.8 applied twice,  $G = S \cup H$  is upward planar. See Figure 6.5(a), and Figure 6.3(c) on page 38.

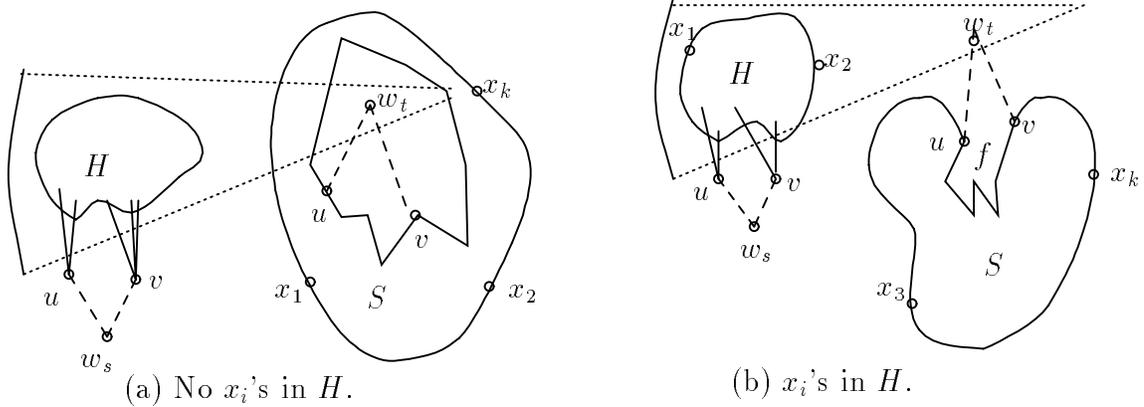


Figure 6.5: Merging of  $S$  and  $H$  when cut-set  $\{u, v\}$  is incomparable.

If no  $x_i \in H$ , then this is sufficient for the embedding. Otherwise, consider the operations carried out above. The joining  $S' \cup H'$  obviously preserves the outer face of  $H'$  and all vertices on the outer face of  $H'$  are also on the outer face of  $S' \cup H'$  as  $w$  is on the outer face of  $H'$  by (ii). Other than for the vertex  $w$ , which is not in  $G$ , this face is also preserved by the splitting of  $w$  and the contraction of  $(u, u')$  and  $(v, v')$ . Thus, if all  $x_i \in H'$  are on the outer face of some  $\overline{H'}$ , they are also on the outer face of  $\overline{G}$  generated by these steps. This is illustrated in Figure 6.5(b). ■

## 6.4 The cut-set $\{u, v\}$ , where $u < v$ , $u \neq s$

Here we consider any other vertex cut-sets not involving the source  $s$ . We divide the graph at a vertex cut  $\{u, v\}$  into two subgraphs—the source component  $S$ , and the remaining subgraph  $H$ . Note that  $v$  can be a source in  $S$ , as long as there is a  $u, v$  path in  $H$ . Again we point out that the markers are required for three purposes—to enforce structural restrictions on the sub-embeddings; to maintain bi-connectivity of the subproblems; and to maintain the single-source property.

**Theorem 6.14** *Let  $G$  be a biconnected directed acyclic graph with a single source  $s$ , and let  $X = \{x_i\} \subseteq V(G)$  be a set of vertices. Let  $\{u, v\}$  be a separation pair of  $G$  with  $u < v$  in  $G$  and  $u \neq s$ , where  $S$  is the source component of  $G$  with respect to  $\{u, v\}$  and  $H$  is the union of all other components. Then,  $G$  admits an upward plane drawing  $\overline{G}$  with all vertices of  $X$  on the outer face if and only if*

- (i)  $S' = (S \cup H\text{-marker})$  admits an upward plane drawing  $\overline{S'}$  with all vertices of  $X$  in  $S$  on the outer face and  $w_t$  (if it exists, otherwise the edge  $(u, v)$ ) on the outer face if some  $x \in X$  is contained in  $H$ .
- (ii)  $H' = (H \cup S\text{-marker})$  admits an upward plane drawing  $\overline{H'}$  with  $w_t$  (if it exists, otherwise the edge  $(u, v)$ ) and all vertices of  $X$  in  $\overline{H}$  on the outer face.

where

$$H\text{-marker} = \begin{cases} M_t & \text{if } v \text{ is a source in } H \\ M_{uv} & \text{if } v \text{ is a sink in } H \\ M_{uvt} & \text{otherwise.} \end{cases}$$

and

$$S\text{-marker} = \begin{cases} M_t & \text{if } v \text{ is a source in } S \\ M_{uv} & \text{otherwise.} \end{cases}$$

**Proof:** (Necessity) Suppose  $G$  admits an upward plane drawing with all  $x_i \in X$  on the outer face.

(Necessity of condition (i)): If  $v$  is a source in  $H$ , then there exists some  $w_t$  in  $H$  and vertex disjoint paths  $u \xrightarrow{\pm} w_t$  and  $v \xrightarrow{\pm} w_t$  by Proposition 6.3; so  $S' = S \cup M_t$  is homeomorphic to a subgraph of  $G$  and is upward planar. If  $v$  is a sink in  $H$ , then  $u$  is the single source of  $H$ , as only  $u$  and  $v$  are possible sources. Thus, in  $H$ , there is a path  $u \xrightarrow{\pm} v$ , so  $S' = S \cup M_{uv}$  is homeomorphic to a subgraph of  $G$  and is upward planar. If  $v$  is neither a source nor a sink in  $H$  then, by Lemma 6.5, there is also some  $w_t > v$  and disjoint directed paths  $u \xrightarrow{\pm} w_t$  and  $v \xrightarrow{\pm} w_t$  in  $G$ . Since  $v$  is a non-source in  $H$ , there is also a  $u \xrightarrow{\pm} v$  path in  $H$ . This path crosses the  $u \xrightarrow{\pm} w_t$  path at some latest vertex  $z$  on that path, so  $S \cup (u \xrightarrow{*} z) \cup (z \xrightarrow{\pm} v) \cup (z \xrightarrow{\pm} w_t) \cup (v \xrightarrow{\pm} w_t)$  is a subgraph of  $G$  and hence upward planar. Note that these four paths are disjoint. Since  $z$  has in-degree one we can contract the  $u \xrightarrow{*} z$  path to  $u$  without destroying upward planarity, by Lemma 6.8, so  $S \cup \{(u, v), (u, w_t), (v, w_t)\}$  has an upward planar subdivision and is upward planar itself. No other vertices of this graph can lie inside the  $u, v, w_t$  triangle, as  $s < u$  (and hence below/outside the triangle) and there are no other  $(u, v)$  components in  $S'$  (as we chose  $S$  to be the *single* source component), so the extra edges and vertex for  $M_{uvt}$  can be added without destroying planarity<sup>1</sup>.

(Necessity of condition (ii)): If  $v$  is a source in  $S$ , then, by Proposition 6.3, there are vertex disjoint paths  $s \xrightarrow{\pm} w_t$  and  $v \xrightarrow{\pm} w_t$  in  $S$ . There must be an  $s \xrightarrow{\pm} u$  path in  $S$ , otherwise there is either a second source ( $u$  is a source in  $H$ , so it cannot also be a source in  $S$ ) or a cycle in  $G$  ( $u < v$  in  $G$ , so there can be no  $v \xrightarrow{\pm} u$  directed path in  $S$ ). Let  $z$  be the last vertex common to paths  $s \xrightarrow{\pm} u$  and  $s \xrightarrow{\pm} w_t$ . Then,  $H \cup \{(z, u), (z, w_t), (v, w_t)\}$  is homeomorphic to a subgraph of  $G$  and is upward

---

<sup>1</sup>The point of adding these edges is to fix the face in  $S$  for the sufficiency conditions.

planar. Since  $\deg^-u = 1$  (in this graph), the edge  $(z, u)$  can be contracted without destroying upward planarity, by Lemma 6.8, and  $H' = H \cup M_t$  is upward planar.

Otherwise ( $v$  a non-source), if  $u < v$  in  $S$ , then  $H' = H \cup M_{uv}$  is homeomorphic to a subgraph of  $G$  and, hence, is upward planar. If  $u$  and  $v$  are incomparable in  $S$ , then they share a greatest lower bound  $w_s$ , by Lemma 6.4, and  $H \cup \{(w_s, u), (w_s, v)\}$  is upward planar. Again,  $\deg^-u = 1$  in  $H$ , so the  $(w_s, u)$  edge can be contracted to give  $H' = H \cup M_{uv}$ .

The necessity of the outer facial conditions on the  $x_i$ 's is similar to the proof of Theorem 6.13.

(Sufficiency) Suppose  $S'$  and  $H'$  admit upward plane drawings  $\overline{S'}$  and  $\overline{H'}$  with all  $x_i$ 's of  $X$  on the respective outer faces,  $v$  on the outer face of  $\overline{H'}$ , and  $w_t$  on the outer face of  $\overline{S'}$  if some  $x_i$  in  $X$  is contained in  $H$ . Consider first the case when no  $x_i$ 's are contained in  $H$ .

**Case 1:**  $v$  is a source in  $H$ : If  $v$  is a source in  $H$  it cannot at the same time be a source in  $S$  as  $u < v$  in either  $S$  or  $H$ . Since  $H'$  is upward planar with a single source  $u$ , and  $(u, v)$  on an outer face,  $H'' = H' \cup \{(w_s, u), (w_s, v)\}$  is upward planar by Lemma 6.11.  $H''$  can be added above sink  $w_t$ , so  $S' \cup H''$  with  $w_s$  and  $w_t$  identified as the same vertex  $w$  is upward planar. Since  $w$  is of in-degree two and out-degree two, it can be split into left and right edges and these can be contracted, by Lemma 6.8, to give an upward planar graph which contains  $G$  as a subgraph, so  $G$  is upward planar; see Figure 6.6(a).

**Case 2:**  $v$  is a sink in  $H$ : If  $v$  is a non-source in  $S$ , then  $H' = H \cup (u, v)$  is upward planar with  $u$  and  $v$  on the outer face by assumption. Then, in  $\overline{S'}$ , there exists an  $\varepsilon$ -ball around some point on the embedded edge  $(u, v)$  into which  $\overline{H'}$  (with  $u$  and  $v$  renamed as  $u'$  and  $v'$ ) can be embedded (Lemma 6.10), and  $(u', v')$  removed. Since

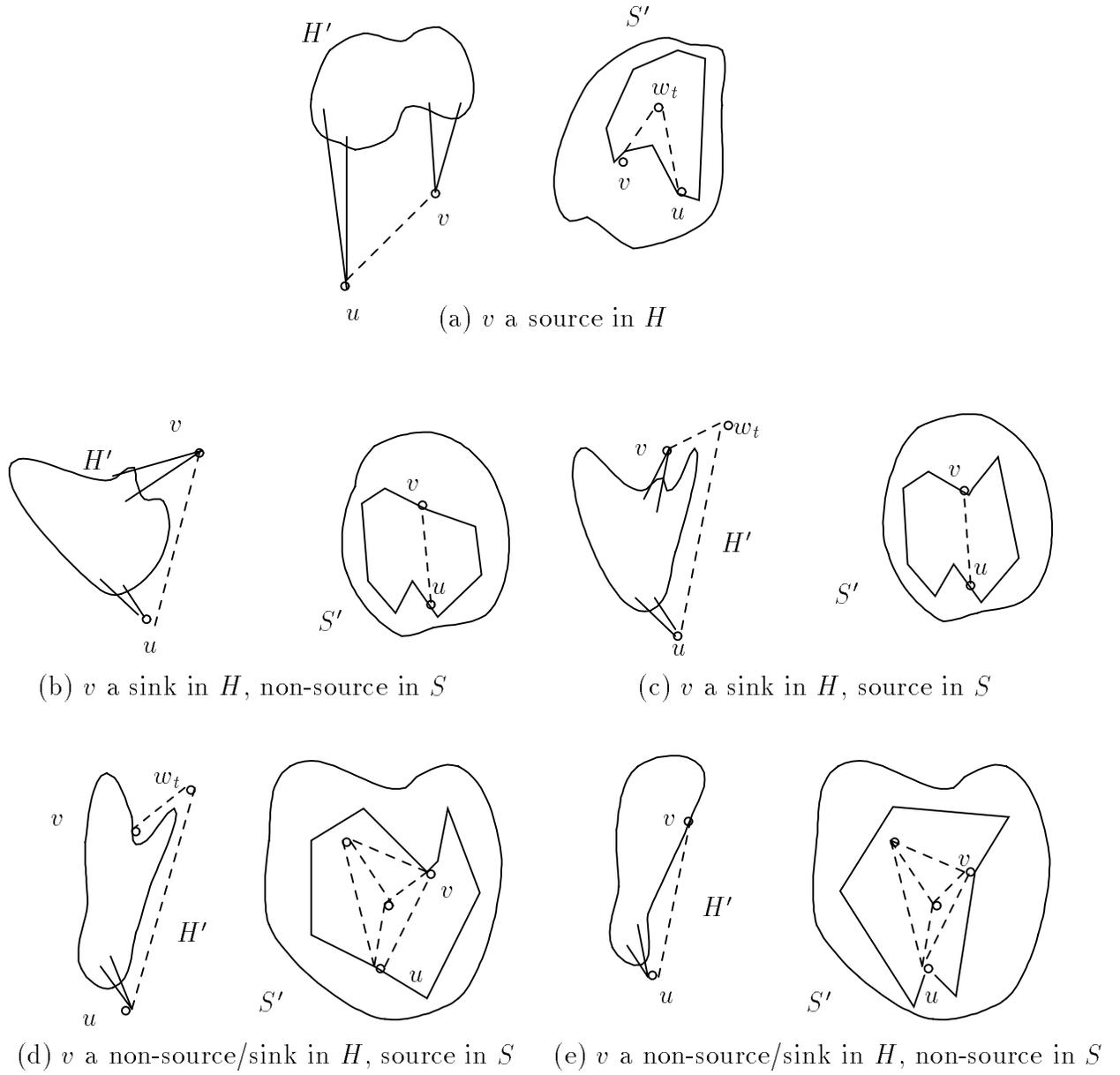


Figure 6.6: Merging of  $S$  and  $H$  with cut-set  $\{u, v\}$  and  $u < v$ .

$u'$  and  $v'$  have in-degree and out-degree one respectively, the extra edges  $(u, u')$  and  $(v, v')$  can be contracted, by Lemma 6.8, and the result  $G$  is upward planar. If  $v$  is a source in  $S$ ,  $\deg^+ v = 1$  in  $H' = H \cup M_t$  so we first apply Lemma 6.8 to contract the edge  $(v, w_t)$  and the the result  $G$  is upward planar. These two possibilities are illustrated in Figure 6.6 (b) and (c).

**Case 3:**  $v$  is a non-source/sink in  $H$ : Suppose  $v$  is a source in  $S$ . Then  $w_t$  is a sink on the outer face of some embedding  $\overline{H'}$  of  $H'$  by condition (ii) of the theorem.  $S'$  with  $H'$  inserted into the marker edge  $(u, v)$  (with  $u, v$  in  $H'$  renamed to  $u', v'$ ), and attached at  $u'$  and  $w_t$  is upward planar by Lemma 6.10. The subgraph with all but  $(u, v)$  of the  $H$ -marker is thus upward planar. Now  $u$  and  $v'$  have in-degree one, so the edges  $(u, u')$  and  $(v, v')$  can be contracted, by Lemma 6.8. Similarly  $w_t$  now has in-degree one, so  $(v', w_t)$  can also be contracted and  $G$  is upward planar; see Figure 6.6(d).

Suppose then that  $v$  is a non-source in  $S$ . The reasoning above applies except for that we cannot immediately contract the  $(v', v)$  edge (since  $v'$  no longer has in-degree one). Now, the edge  $(u', v')$  lies between  $u$  and  $v$  and the the fact that  $u$  is the single source of  $H'$  ensure that  $H'$  can be scaled to some small area in the left half-plane beside  $v'$  so that it requires some part of the upper half-plane above  $v$ . Some such region must exist in the face  $f$  of  $S'$  containing the marker, since  $S'$  is upward planar with the dummy edge above  $v$ . Thus with  $H$  appropriately scaled before the inserting it into the  $(u, v)$  edge, the edge  $(v, v')$  can be contracted without destroying upward planarity.

Again, the conditions on the  $x_i$ 's follow in a similar manner to those in the previous section, as they simply involve the choice of the outer face of  $H \cup S$ . ■

## 6.5 The cut-set $\{s, v\}$

As mentioned in the introduction to this chapter, it is important to be able to identify the component which is going to be the “outer” one—that will contain the other component. The “inner” component will have to have its marker on the outer face since this is the structure it will be a proxy for. If this property is not known, we have to check the subproblems subject to both cases, and an exponential time blowup results.

Until now, this component has been uniquely identified as the *source component*, since that component cannot lie within an internal face of any other component. This provided the needed structural restrictions for the embedding—in that  $H$  can lie inside  $S$ , but  $S$  can never lie inside  $H$ . If the cut-set we wish to use contains  $s$  as one of the cut-set vertices, we lose this restriction, so we handle it instead by choosing a 3-connected component as an “identified” component, to avoid the exponential blowup that results from searching for the “outside” component. The reason that this works is because a triconnected component can be checked for upward planarity without any further recursive calls to find triconnected components, and can be checked in linear time by algorithm S-E-UP-PLANAR of Chapter 3—since the restriction for  $s$  and  $v$  on the outer face will fix the outer face, only one possible embedding that needs to be checked exists. If  $G$  has only cut-sets of the form  $\{s, v\}$ , then, for at least one such cut-set, a triconnected component exists. In linear time, given the list of cut-sets, this component can be found by depth-first search.

We state this result in terms of two theorems. One is applicable if the triconnected component can be embedded within the other component, and one when it cannot.

**Theorem 6.15** *Let  $G$  be a biconnected DAG with a single source  $s$ , and let  $X = \{x_i\} \subseteq V(G)$  be a set of vertices. Let  $\{s, v\}$  be a separation pair of  $G$ ,  $E$  be a 3-connected component of  $G$  with respect to  $\{s, v\}$ , and  $F$  be the union of all other components of  $G$  with respect to  $\{s, v\}$ . If  $E$  admits an upward plane drawing with  $s$  and  $v$  on the outer face, then  $G$  admits an upward plane drawing with all vertices of  $X$  on the outer face if and only if*

- (i)  $F' = (F \cup E\text{-marker})$  admits an upward plane drawing  $\overline{F'}$  with all vertices of  $X$  in  $F$  on the outer face, and  $w_t$  (if it exists, otherwise the edge  $(u, v)$ ) also on the outer face if some  $x \in X$  contained in  $E$ .
- (ii)  $E' = (E \cup F\text{-marker})$  admits an upward plane drawing  $\overline{E'}$  with  $w_t$  (if it exists, otherwise the edge  $(u, v)$ ) and all vertices of  $X$  in  $E$  on the outer face,

where

$$E\text{-marker} = \begin{cases} M_t & \text{if } v \text{ is a source in } E \\ M_{uv} & \text{if } v \text{ is a sink in } E \\ M_{uvt} & \text{otherwise.} \end{cases}$$

and

$$F\text{-marker} = \begin{cases} M_t & \text{if } v \text{ is a source in } F \\ M_{uv} & \text{otherwise.} \end{cases}$$

**Proof:** The similarity of this theorem to Theorem 6.14 of the previous section should be evident; the only major difference is the way in which the outer component is isolated; as such, we will be more terse in its proof. For compatibility with previous notation, we will continue to use  $u$  and  $v$  as the cut-vertices, even though  $u = s$ .

(Necessity) Assume that  $G$  has an upward plane embedding  $\overline{G}$  with all vertices of  $X$  on the outer face, and furthermore that  $E$  admits an upward plane embedding  $\overline{E}$  with  $v$  on the outer face.

(Necessity of condition (i)): If  $v$  is a source in  $E$ , then there exist vertex disjoint directed paths  $u \xrightarrow{\pm} w_t$  and  $v \xrightarrow{\pm} w_t$  for some  $w_t$  in  $E$  (Proposition 6.3) and  $F' = F \cup M_t$  is upward planar. If  $v$  is a sink in  $E$ , then there is a directed path  $u \xrightarrow{\pm} v$  in  $E$  and  $F' = F \cup M_{uv}$  is upward planar. Otherwise,  $\deg^+ v \geq 1$  and there exists some vertices  $w_t$  and  $z$  in  $E$  such that  $u \xrightarrow{*} z$ ,  $z \xrightarrow{\pm} v$ ,  $v \xrightarrow{\pm} w_t$  and  $z \xrightarrow{\pm} w_t$  are vertex disjoint directed paths in  $E$  (Lemma 6.5) and  $F \cup \{(u, v), (v, w_t), (u, w_t)\}$  is homeomorphic to a subgraph of an upward planar graph,  $G$ , and is upward planar. Within this triangle in any upward plane embedding, we can add the additional “dummy” vertex and  $F' = F \cup M_{uvt}$  is also upward planar. If some  $x \in X$  is on the outer face of  $G$ , it is certainly on the outer face of the sub-embedding  $F'$ .

(Necessity of condition (ii)): If  $v$  is a source in  $F$ , then there exist vertex disjoint directed paths  $u \xrightarrow{\pm} w_t$  and  $v \xrightarrow{\pm} w_t$  for some  $w_t$  in  $F$  (Proposition 6.3) and  $E' = E \cup M_t$  is upward planar. Otherwise, we note that  $u < v$  in  $G$ , since  $u = s$  is the single source of  $G$ , so  $E' = E \cup M_{uv}$  is homeomorphic to a subgraph of  $G$  and is upward planar.

Note that the necessity of the outer facial conditions again follows from Theorem 6.5.

(Sufficiency) Suppose  $G$ ,  $E$  and  $F$  are as defined, and conditions (i) and (ii) of the theorem are met.

Consider first when no  $x \in X$  is contained in  $E$ .

**Case 1:**  $v$  is a source in  $E$ : If  $v$  is a source in  $E$  it cannot at the same time be a source in  $F$  as  $u < v$  in either  $E$  or  $F$ . Since  $E'$  is upward planar with a single source  $u$ , and  $(u, v)$  on an outer face,  $E'' = E' \cup \{(w_s, u), (w_s, v)\}$  is upward planar. By Lemma 6.11,  $E''$  can be added above sink  $w_t$  in  $F'$ , so  $F' \cup E''$  with  $w_s$  and  $w_t$  identified as the same vertex  $w$  is upward planar. By application of Lemmas 6.12

and 6.8,  $w$  can be split, and the resulting edges contracted, resulting in an upward planar graph which contains  $G$  as a subgraph, so  $G$  is upward planar.

**Case 2:**  $v$  is a sink in  $E$ : If  $v$  is a non-source in  $F$ , then  $E' = E \cup (u, v)$  is upward planar with  $(u, v)$  on the outer face by condition (ii) of the theorem. By Lemmas 6.10 and 6.8 an embedding of  $E'$  (with  $u, v$  renamed to  $u', v'$ ) can be inserted into the edge  $(u, v)$  of  $F$  (attached at  $u'$  and  $v'$ ) the edge  $(u', v')$  removed, the other extra edges contracted, and the result  $G$  is upward planar. If  $v$  is a source in  $F$ ,  $\deg^+ v = 1$  in  $E' = E \cup M_t$ , so we first apply Lemma 6.8 to contract the edge  $(v, w_t)$ , and the result follows.

**Case 3:**  $v$  is a non-source/sink in  $E$ : Suppose  $v$  is a source in  $F$ . Then  $w_t$  is a sink on the outer face of some embedding  $\overline{E'}$  of  $E'$  by condition (ii) of the theorem.  $F'$  with  $E'$  inserted into the marker edge  $(u, v)$  (with  $u$  and  $v$  in  $E'$  renamed to  $u'$  and  $v'$ ) and attached at  $u'$  and  $w_t$  is upward planar by Lemma 6.10. The subgraph with all but  $(u, v)$  of the  $E$ -marker is thus upward planar. Now  $u$  and  $v'$  have in-degree one, so the edges  $(u, u')$  and  $(v, v')$  can then be contracted by Lemma 6.8. Similarly  $w_t$  now has in-degree one, so  $(v', w_t)$  can also be contracted, and  $G$  is upward planar.

Suppose then that  $v$  is a non-source in  $F$ . The reasoning above applies except for the ability to immediately contract the  $(v', v)$  edge (since  $v'$  no longer has in-degree one). Now, the edge  $(u', v')$  between  $u$  and  $v$  and the the fact that  $u$  is the single source of  $E'$  ensures that  $E'$  can be scaled to some small area in the left half-plane beside  $v'$  and requires some part of the upper half-plane above  $v$ . Some such region must exist in the face  $f$  of  $F'$  containing the marker, since  $F'$  is upward planar with the “dummy” edge above  $v$ . Thus with  $E$  appropriately scaled before its insertion into the  $(u, v)$  edge, the edge  $(v, v')$  can be contracted without destroying upward planarity. ■

**Theorem 6.16** *Let  $G$  be a biconnected DAG with a single source  $s$  and let  $X = \{x_i\} \subseteq V(G)$  be a set of vertices. Let  $\{s, v\}$  be a separation pair of  $G$ ,  $E$  be a 3-connected component of  $G$  with respect to  $\{s, v\}$ , and  $F$  be the union of all other components of  $G$  with respect to  $\{s, v\}$ . If  $E$  does not admit an upward plane drawing with  $s$  and  $v$  on the outer face, then  $G$  admits an upward plane drawing with all vertices of  $X$  on the outer face if and only if*

- (i) *There is no  $x \in X$  contained in  $F$ .*
- (ii)  *$F' = (F \cup E\text{-marker})$  admits an upward plane drawing  $\overline{F'}$  with  $w_t$  (if it exists, otherwise the edge  $(u, v)$ ) also on the outer face if some  $x \in X$  is contained in  $E$ .*
- (iii)  *$E' = (E \cup F\text{-marker})$  admits an upward plane drawing  $\overline{E'}$  with all  $x \in X$  on the outer face,*

where

$$E\text{-marker} = \begin{cases} M_t & \text{if } v \text{ is a source in } E \\ M_{uv} & \text{if } v \text{ is a sink in } E \\ M_{uv} & \text{otherwise.} \end{cases}$$

and

$$F\text{-marker} = \begin{cases} M_t & \text{if } v \text{ is a source in } F \\ M_{uv} & \text{if } v \text{ is a sink in } F \\ M_{uvt} & \text{otherwise.} \end{cases}$$

**Proof:** If  $E$  has no upward plane drawing with  $u$  and  $v$  both on the outer face, then the only way  $G$  could be upward planar is if  $F$  can be embedded within a face of  $E$ . Thus, the outer face of  $G$  is fixed as being some face of  $\overline{E'}$  not containing  $v$  or any vertices of  $F$ , or some face which can be partitioned into vertices from the outer face of  $F$  and the outer face of  $G$ . It remains to ensure that there is

some embedding of  $F$  which will fit the structural constraints of the shape of a face shared by  $u$  and  $v$  in  $\overline{E}$ . These are exactly the conditions previously required by  $E$  for embedding within  $\overline{F}$ . The proof is identical to Theorem 6.15 up to the naming of  $E$  and  $F$ . The differences in the statement of the theorems affect only the time bound. The result is identical, as it does not depend upon triconnectedness of either component. ■

## Chapter 7

# An Algorithm for Upward Planar Embedding

The algorithm works as follows: A directed acyclic graph  $G$  with a single source  $s$  is given to BI-MERGE, which splits  $G$  into biconnected components, calls CONTRACT which contracts subdivision edges, identifies the triconnected components in the result, and calls TRI-MERGE to split the result into triconnected components. TRI-MERGE calls itself recursively to isolate triconnected components, then calls TRI-EMBED to test all of the embeddings of a triconnected component for upward planarity with MOD-S-E-UP-PLANAR (modified slightly from S-E-UP-PLANAR of Chapter 4 to take account of markers), returning an embedding. As the embeddings are returned to the calling routines, CONTRACT un-contracts the edges and TRI-MERGE merges the embeddings of the recursively defined sub-graphs, as does BI-MERGE. If any violations of upward planarity are found, the algorithm rejects.

**Algorithm BI-MERGE:** Find the set of cut-vertices, “CutList”, and the set

of biconnected components, “ComponentList”, of the input graph  $G$ . Find an upward planar embedding (if one exists) of each component in ComponentList with the CONTRACT algorithm. Then, for each cut vertex  $v$ , remove the source component  $S$  and remaining  $v$ -components  $\{H_i\}$  from ComponentList. Determine a face above  $v$  in  $S$  and embed the  $\overline{H_i}$ 's in the prescribed face  $f$ . If  $f$  is not the outer face, add dummy edges from the sinks of each embedding  $H_i$  to the covering vertex of  $v$ , to resolve the internal sinks generated by the  $H_i$ 's (analogous to the more detailed description of Chapter 4). Add the result to ComponentList, and continue until CutList is exhausted.

Finding biconnected components can be done in linear time, and since the merge step considers each cut-vertex at most once and need not search the remainder of the graph, this step requires linear time plus the time taken by CONTRACT on each sub-problem (subgraph).

**Algorithm CONTRACT:**

Perform a depth first search of the input graph  $G$  and contract edges around any vertices of in-degree one and out-degree one, (following Lemma 6.7) storing the required information within the contracted edges. Use the tri-connected components algorithm [19] to find the list of separation pairs of  $G$ . Call TRI-MERGE with the contracted graph to find an upward plane embedding, then subdivide the edges of the embedding with the previously contracted vertices.

Depth first search takes linear time, so this step is also linear time plus the time taken by TRI-MERGE on the initial subproblem.

**Algorithm TRI-MERGE:**

The input is a graph  $G$ , a list of separation pairs  $L$  in  $G$  and a list  $X$  of vertices  $X$  required on the outer face of  $\overline{G}$ . Pick a cut-set  $\{u,v\}$  of  $G$  from the list of cut-sets

$L$ , (choosing all cut-sets not containing  $s$  before those containing  $s$ ) and determine whether  $u$  and  $v$  are comparable in  $G$ . If they are, ensure that  $u < v$  in the labeling. If no such cut-set exists, call TRI-EMBED to find an upward planar representation of  $G$  with the  $x_i$ 's on the outer face, and return the result.

If  $u \neq s$ , isolate the source component  $S$  with a depth-first search, form the union  $H$  of the remaining components (following Theorems 6.13 and 6.14) and add the required marker edges to each component. At the same time, determine the new sets  $X_S$  and  $X_H$ , and  $L_S$  and  $L_H$ , from  $X$  and  $L$  respectively, and recursively call TRI-MERGE with each sub-problem.  $X_S$  and  $X_H$  are a partition of  $X$  into the set of vertices of  $X$  which are also in  $S$  and the set of vertices of  $X$  which are also in  $H$ , plus the additional required vertices which must be on the outer face of the sub-problems, and possible additions for new marker edges.  $L_H$  and  $L_S$  are a partition of  $L$  into the set of cut-sets of  $G$  contained in  $H$  and  $S$  respectively. With the results of each sub-problem, apply the algorithm implicit in the sufficiency proof of Theorems 6.13 or 6.14 (as appropriate) to combine the two embeddings with contraction operations. As in BI-MERGE, if one component is added within an internal face of another, add an edge from each of its sinks to the vertex covering  $u$  in the outer component to resolve the inner sinks. Return the result.

If  $u = s$ , isolate a component  $E$  of  $G$  with respect to some  $\{s, v\}$  which is triconnected, and let  $F$  be the union of all other  $\{s, v\}$ -components. This is possible because we choose 3-connected components to avoid  $s$  when non-source cut-sets exist, and if all cut sets involve one vertex  $s$ , then at least one has a triconnected component. Add the marker edges and determine  $X_E$  and  $X_F$  as above per Theorem 6.15. Call TRI-MERGE with the  $(E', \emptyset, X_E \cup \{u, v\})$  sub-problem, to test whether  $E$  has an upward embedding satisfying the previous conditions and also having  $u$  and  $v$  on the outer face. The cut-set list is empty as  $E$  is triconnected.

If TRI-MERGE passes, then  $E$  can be embedded in a face of  $F$  (possibly the outer face), so compute the subproblem  $(F', L_F, X_F)$ , merge results according to the proof of Theorem 6.15, and resolve the sinks of  $\overline{E}$  which are internal in the result. Reject if the subproblem fails.

If TRI-MERGE fails, then  $F$  cannot have any  $x_i$ 's (by Theorem 6.16); the embedding of  $F$  must lie strictly within an internal face of the embedding of  $E$  (i.e. it is not even possible for  $E$  and  $F$  to “together” form the outer face; the outer face must consist entirely of vertices of  $E$ ), so reject if  $X_F$  is not empty. Call TRI-MERGE with the subproblems  $(F', L_F, \{u, v\})$  and  $(E', \emptyset, X_E)$ , merge the results with the contraction operations of the algorithm implicit in the sufficiency conditions of Theorem 6.16, and resolve the sinks of  $\overline{F}$  which are internal in the result. Again, reject if either subproblem fails.

Depth first search is a linear time operation that can determine all the required information, such as whether  $u$  and  $v$  are comparable and the partitions of  $X$  and  $L$ . The first call to TRI-EMBED in the  $s = u$  case is with a tri-connected graph and thus requires only linear time (time for of S-E-UP-PLANAR). Subsequently, the outer face is chosen, so a constant number of disjoint calls are made to TRI-MERGE (i.e. no backtracking is required to find the outer face). In total, we execute a linear time operation for each cut-set, of which there are no more than  $n$ , followed by an  $O(n^2)$  time operation (TRI-EMBED) on some number of subgraphs of  $G$ . The total time is thus quadratic.

**Algorithm TRI-EMBED:**

Compute the faces of the tri-connected input graph  $G$  and a planar embedding of  $G$  using a linear time algorithm [18, 4].

For each face  $f$  adjacent to the source, transform the abstract embedding  $\overline{G}$

of  $G$  to have outer face  $f$  and test for strongly equivalent upward planarity with Algorithm MOD-S-E-UP-PLANAR (below). The transformation is simply a matter of choosing a different outer face in the specification of the abstract embedding. If the list of faces is exhausted, abort, otherwise return the first valid embedding. Note that if we have more than just the source (i.e. another vertex or an edge) specified to be on the outer face, then there are no more than a constant number of faces to check.

This step requires a linear number of calls to MOD-S-E-UP-PLANAR, since a planar triconnected graph has a linear number of possible outer faces [3]. Specifically, it requires  $deg^+$  calls if no other vertices are specified on the outer face and at most two calls otherwise.

**Algorithm MOD-S-E-UP-PLANAR:**

Perform the algorithm of S-E-UP-PLANAR, with the following modification:

Whenever a distinguished edge is uncovered, expand it into markers and edges with all  $M_t$  markers outside of all  $M_{uvt}$  markers, all  $M_{uvt}$  markers outside all  $M_{uv}$  markers, and all  $M_{uv}$  markers outside all real edges  $(u, v)$  of  $G$  which are in turn outside all  $M_s$  markers, then continue as before.

This step requires only local changes and the time is still proportional to the number of marker components hidden in the given edge, which is linear over the total time of the algorithm. Thus MOD-S-E-UP-PLANAR runs in linear time.

## Chapter 8

# Drawing Upward Planar Graphs

Here we wish to address the issue of constructing a physical drawing from the abstract representation constructed thus far. This is straightforward, since the output of the algorithm of Chapter 7 is an upward planar embedding  $\overline{G}$  of  $G$  (with all internal sinks resolved by extra edges). All sinks are on the outer face, so a new vertex  $t$  can be added to the outer face and connected to each sink of  $\overline{G}$  and to  $s$  (also on the outer face if  $G$  is to be upward planar). The result is then a representation of a planar  $st$ -graph, and we can then apply the known algorithms of either Chiba and Nishizeki [6] or DiBattista and Tomassia [10] to give a drawing (in linear time for a monotonic drawing or “ $O(n \log n)$ ” time for a straight-line drawing), and then remove the dummy edges.

Recall that the straight-line algorithm is output sensitive, and with time measured on this basis, can take possibly exponential time, as discussed in Section 3.4. This is, however, optimal for the output sensitive case.

# Chapter 9

## Conclusions

We have given a linear time algorithm that tests whether a given single source acyclic digraph is upward planar strongly equivalent to a given planar embedding, and gives representation of this drawing if it exists. We have used this result to give an efficient  $O(n^2)$  algorithm to test for upward planarity of a single source acyclic digraph.

A lower bound for the single-source upward planarity problem is not known, although we believe that it may be possible to perform the entire test in sub-quadratic (perhaps linear) time. An obvious extension of this work would be to find such an algorithm or prove a lower bound.

This thesis has concentrated on the issues of efficiently testing for an upward planar embedding and outputting an abstract representation of such an embedding. However, by illustrating how the input graph can be embedded within a planar *st*-graph, we have also shown how to take advantage of known methods for constructing a physical upward plane drawing of the graph.

Although it is not possible to achieve sub-exponential area in drawing an arbi-

trary upward planar graph, it would be an interesting problem to guarantee sub-exponential area where possible and to characterize the class of upward planar graphs requiring exponential grid-size. Guaranteeing minimum area in all cases is, however, NP-hard [29].

The more general problem of testing for an upward planar drawing of an arbitrary acyclic digraph is open. The only known characterization is that any such graph is a subgraph of an upward planar graph with a single source and sink [10].

# Bibliography

- [1] A.V. AHO, J.E. HOPCROFT AND J.D. ULLMAN. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Mass., 1974.
- [2] G. BIRKHOFF. *Lattice Theory*, 3rd ed. Amer. Math. Soc., Providence, R.I., 1967.
- [3] J.A. BONDY AND U.S.R. MURTY. *Graph Theory with Applications*. MacMillan Co. New York, 1976.
- [4] K.S. BOOTH AND G.S. LUEKER. *Testing the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*. J.Comput.Syst.Sci. 13, pp. 335-379, 1976.
- [5] N. CHIBA, T. YAMANOUCHI AND T. NISHIZEKI. *Linear algorithms for convex drawings of planar graphs*, in *Progress in Graph Theory* (J.A. Bondy and U.S.R. Murty eds.) Academic Press, New York. pp. 153-173, 1984.
- [6] N. CHIBA, T. NISHIZEKI, S. ABE AND T. OZAWA. *A linear algorithm for embedding planar graphs using PQ-trees*. J.Comput.Syst.Sci. **30,1**, pp. 54-76, 1985.
- [7] M. CHROBAK. *A linear-time algorithm for drawing a planar graph on a grid*. Manuscript, Univ.Calif.Riverside, 1988.

- [8] H. DEFRAYSSEIX, J. PACH, AND R. POLLACK. *Small sets supporting Fary embeddings of planar graphs*. Proc. 20'th Annual Symposium on the Theory of Computing, pp. 426-433, 1988.
- [9] G. DIBATTISTA, W. LIU AND I. RIVAL. *Bipartite graphs, upward drawings, and planarity*. Università Degli Studi di Roma, "La Sapienze", TR RAP.05.90 (1990).
- [10] G. DIBATTISTA AND R. TOMASSIA. *Algorithms for plane representations of acyclic digraphs*. Theoretical Computer Science 61, pp. 175-178, 1988.
- [11] G. DIBATTISTA AND E. NARDELLI. *An algorithm for testing planarity of hierarichical graphs*, in *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science 246. (Springer, Berlin), pp. 277-289, 1987.
- [12] G. DIBATTISTA, R. TOMASSIA AND I.G. TOLLIS. *Area requirement and symmetry display in drawing graphs*. Proc. ACM Symposium on Computational Geometry, pp. 51-60, 1989.
- [13] P. EADES, X. LIN, AND R. TAMASSIA. *A new approach for drawing a hierarchical graph*. Proc. Second Canadian Conference on Computational Geometry, J. Urrutia ed., August 1990.
- [14] S. EVEN. *Graph Algorithms*. Computer Science Press, Rockville, Maryland, 1979.
- [15] S. EVEN AND R. TARJAN. *Computing an st-numbering*, Theoretical Computer Science 2, pp. 339-344, 1976.
- [16] I. FÁRY. *On straight line representing of planar graphs*. Acta. Sci. Math. (Szeged) 11, pp. 319-321, 1948.

- [17] S. FOLDES, I. RIVAL AND J. URRUTIA. *Light sources, obstructions and spherical orders*. University of Ottawa TR-88-32, August 1988.
- [18] J. HOPCROFT AND R. TARJAN. *Efficient planarity testing*. J.ACM 21,4, pp. 549-568, 1974.
- [19] H. HOPCROFT AND R.E. TARJAN. *Dividing a graph into triconnected components*. SIAM J. Comput. 2, pp. 135-158, 1972.
- [20] D. KELLY. *Fundamentals of planar ordered sets*. Discrete Math 63, pp. 197-216, 1987.
- [21] D. KELLY AND I. RIVAL. *Planar lattices*. Can. J. Math., Vol XXVII, No. 3, pp. 636-665, 1975.
- [22] A. LEMPEL, S. EVEN, AND I. CEDERBAUM. *An algorithm for planarity testing of graphs*, in *Theory of Graphs, International Symposium, Rome*, P. Rosenstiehl, Ed., Gordon and Breach, N.Y. pp. 215-232, (1967).
- [23] T. NISHIZEKI AND N. CHIBA. *Planar Graphs: Theory and Algorithms*. Annals of Discrete Mathematics 32, Elsevier Science Publishing Company, Inc. New York, 1988.
- [24] C.R. PLATT. *Planar lattices and planar graphs*. J. Comb. Theory (B) 21, pp. 30-39, 1976.
- [25] R. READ. *New methods for drawing a planar graph given the cyclic order*. Congressus Numerantium Vol. 56, pp. 31-44, 1976.
- [26] I. RIVAL. *Order, invariance and visibility*. University of Ottawa TR-89-47, November 1989.

- [27] W. SCHNYDER. *Embedding planar graphs on the grid*. Proc. First ACM-SIAM Symposium on Discrete Algorithms, pp. 138-148, 1990.
- [28] R.E. TARJAN. *Depth-first search and linear graph algorithms*. SIAM J. Comput. 1,2, pp. 146-159, 1972.
- [29] R. TAMASSIA AND P. EADES. *Algorithms for drawing graphs: an annotated bibliography*. Brown University TR CS-89-09, 1989.
- [30] C. THOMASSEN. *Planar acyclic oriented graphs*. Order 5, pp. 349-361, 1987.
- [31] C. THOMASSEN, *Planarity and duality of finite and infinite graphs*. J. Combin. Theory B 29, pp. 244-271, 1980.
- [32] W.T. TROTTER. *Graphs and partially ordered sets*, in *Selected Topics in Graph Theory 2* (L.W. Beineke and R.J. Wilson, editors), Academic Press, New York, pp. 237-268, 1983.
- [33] W.T. TUTTE. *Convex representations of graphs*. Proc. London Math. Soc. Vol.10, pp. 304-320, 1960.
- [34] W.T. TUTTE. *How to draw a graph*. Proc. London Math. Soc. Vol.3, No.13, pp. 743-768, 1963.